Historic, archived document

Do not assume content reflects current
scientific knowledge, policies, or practices.

Contract No.: 53-3198-3-038
MPR Reference No.: 8156-004

TECHNICAL DESCRIPTION OF
THE 1989 AND 1990
HEALTHY EATING INDEX FILES

December 30, 1994

Authors:

James Ohls
Karen Pence

Submitted to:

U.S. Department of Agriculture
Food and Nutrition Service
3101 Park Center Drive, 2nd Floor
Alexandria, VA 22302

Project Officer:
Steve Carlson

Submitted by:

Mathematica Policy Research, Inc.
600 Maryland Avenue, S.W.
Suite 550
Washington, DC 20024

Project Director:
Carole Trippe

CONTENTS

FIGURES

INTRODUCTION

This report provides technical documentation for the files used in creating the Healthy Eating Index. It is intended as a comprehensive guide for programmers who are replicating or building upon our analysis. Our companion document, "The Healthy Eating Index," (Kennedy et al., 1994) explains the conceptual basis of the index.

The following text gives a general descriptive overview of the files, programs, and general file creation process. Appendix A provides detailed technical information for each input file and program. Appendix B shows the contents of the SAS Healthy Eating Index (HEI) analysis file, CSFINDEX.SSD. This output includes the name, length, and description of each variable. In Appendix C, we provide the complete FORTRAN code for CSF_IDX.FOR, which creates the Healthy Eating Index. To illustrate how to use the SAS HEI file, we provide three sample SAS programs, and their output, in Appendix D.

A. DESCRIPTION OF THE FILE

The main source of dietary intake data used was the Continuing Survey of Food Intake by Individuals (CFII) which is conducted periodically, by the U.S. Department of Agriculture. The raw CSFII is a hierarchial file with household, person, nutrition, food item, and dietary awareness records. The Healthy Eating Index (HEI) file is a rectangular SAS file that contains only person records, with relevant household and food information appended. Since persons without three complete days of food intake data, infants, and pregnant and lactating women are excluded from the HEI file, the HEI file contains fewer persons than the raw CSFII.

There are 3,997 observations on the 1989 HEI file, and 3,466 observations on the 1990 file. The files have identical layouts, with 176 variables and a 1396-byte record length.

1

B. VARIABLES

The Healthy Eating Index file contains variables taken directly from the CSFII and variables computed especially for the Healthy Eating Index (HEI). When possible, we retained the CSFII variable name on the HEI file. For complete information about variables from the CSFII, consult the CSFII documentation provided by the U.S. Department of Agriculture. For the algorithms used for computed variables, check Appendix C, which contains the program used to compute the HEI, and Kennedy et al., 1994, which describes the conceptual basis for several of the computed variables. Short descriptions of each variable are provided in Appendix B.

C. WEIGHTING

There are five different weighting variables on the CSFII. Since our unit of analysis is the person, we used WGT_3DAY, which represents persons with three days of dietary intake. WGT_3DAY sums to the total number of persons, in thousands, residing in the 48 contiguous states. The weights also sum correctly to the number of women age 20 or older, the number of men age 20 or older, and the number of persons under age 20. WGT_3DAY is the only weight variable retained on the HEI file.

D. CONSTRUCTION OF THE FILE

The SAS HEI file is the end product of a complex process that includes input files from USDA, MPR's subcontractor, Technical Assessment Systems (TAS), and MPR. It draws on computer programs written in FORTRAN, SAS, and DOS batch language. The following text and Figures 1 and 2 provide an overview of this process. Detailed technical information for the input files and computer programs is provided in Appendix A.

The data source for people on the HEI file is the CSFII. One record on the 1989 CSFII contains an error, which we corrected using the FIXDATA.FOR program. There is no equivalent

FIGURE 1

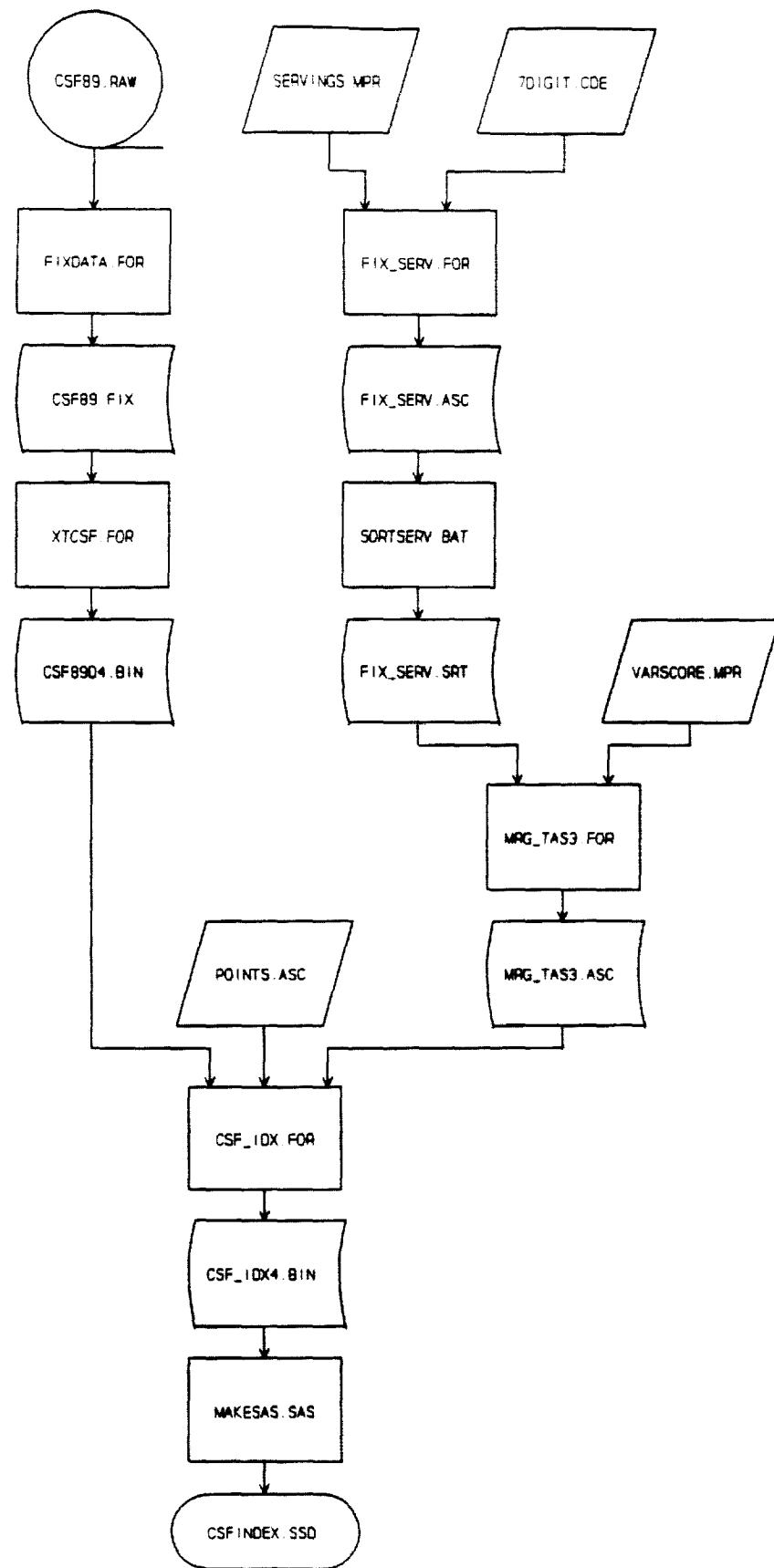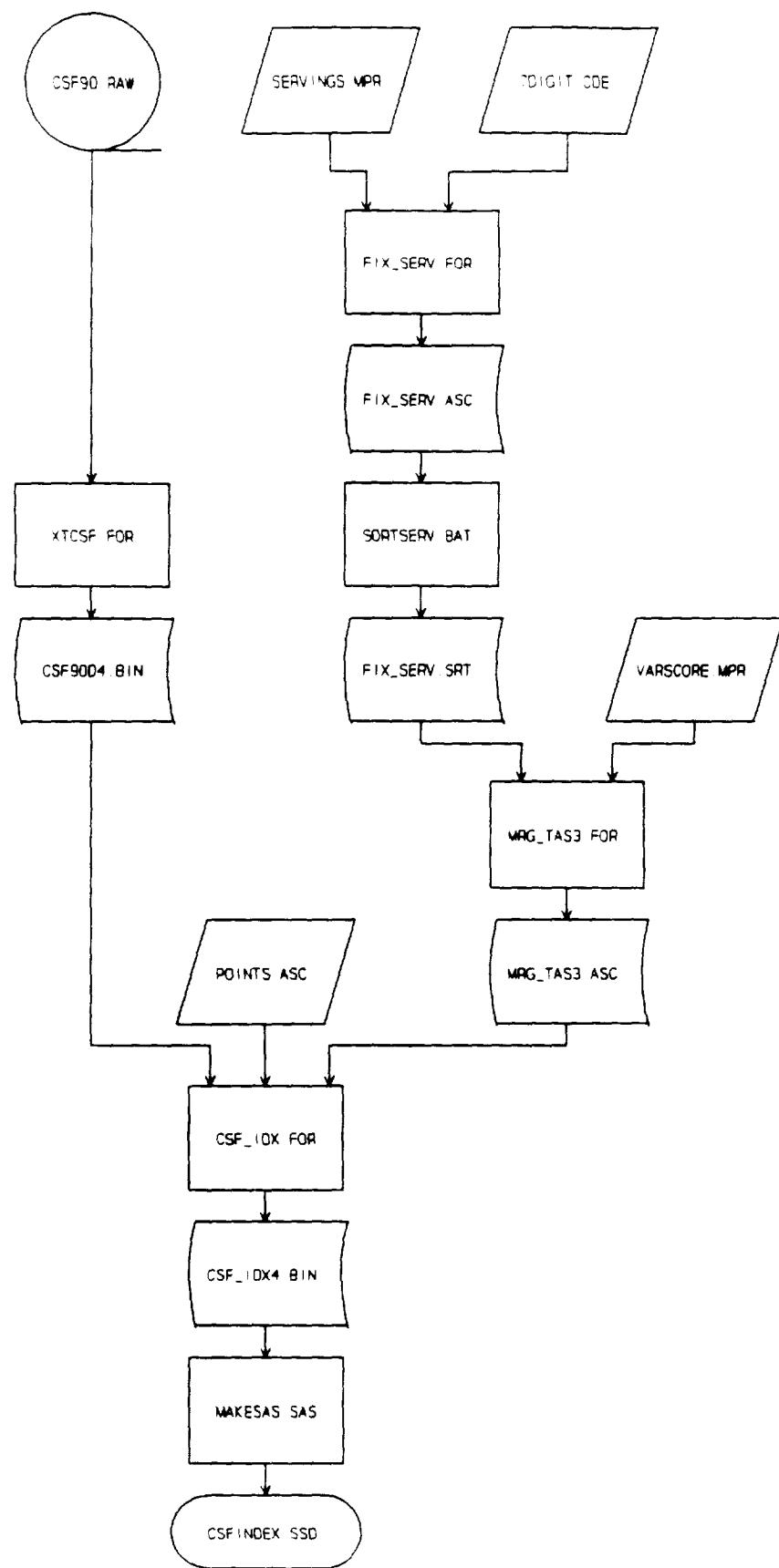## FLOW CHART FOR 1989 FILE

FIGURE 2

FLOW CHART FOR 1990 FILE

error on the 1990 CSFII. Then we created a CSFII extract (CSF89D4.BIN) with complete demographic and dietary information for each person, using the XTCSF.FOR program.

Meanwhile, we created a food-level file which contained nutrition information from three sources. Each observation in this file represented one of the 7-digit USDA food codes. First we merged the labels for these codes (7DIGIT.CDE) with the number of food group servings represented by 100 grams of the food. Using a sorting utility, we sorted the resulting output file, FIX_SERV.ASC, in ascending order by food group code. We merged the ensuing file with the constituent codes for each food. Constituent codes are the major ingredients composing a "mixture" food. For example, beef and bread might be the constituent codes for a hamburger. This information helped us quantify the variety in an individual's diet. After these merges, we had a food-level file (MRG_TAS3.ASC) with enough nutritional information to implement the Healthy Diet Index specifications.

Using the person-level information on CSF89D4.BIN, the food-level information on MRG_TAS3.ASC, and the maximum and minimum point HEI component matrix in POINTS.ASC, we calculated the Healthy Eating Index value for every person in the file and created a new output file, CSF_IDX4.BIN. The program that creates this analysis file, CSF_IDX.FOR, is presented in Appendix C. For more detailed information on the HEI creation algorithms, consult Kennedy et al., 1994. We used this file and its SAS version, CSFINDEX.SSD, to perform tabulations and analyses on the Healthy Eating Index. Three sample analysis programs written in SAS are shown in Appendix D.

E. PROGRAM AND FILE ARCHIVING

The programs and input files documented in Appendix A are archived on a 90 megabyte bernoulli disk. The bernoulli disk includes all of the files used by the FORTRAN programs. All of the files used to create the 1989 Healthy Eating Index can be found in the HEI89 directory. The 1990 files are in the HEI90 directory.

We provide zipped copies of the raw 1989 and 1990 CSFII files. We have changed one record on the 1989 file. Therefore, the raw files have different names: the 1989 file is called CSF89.FIX, and the 1990 file is called CSF90.RAW. These files can be unzipped with PKUNZIP, a shareware utility distributed by PKWARE, Inc. To receive a copy of the software and its documentation, send $47 to PKWARE, 9025 N. Deerwood Drive, Brown Deer, WI 53223.

We also include both SAS and binary versions of our final Healthy Eating Index analysis file. The SAS file is called CSFINDEX.SSD; the binary version is CSF_IDX4.BIN. Although we do not provide explicit documentation for the binary file in this report, its layout can be inferred from both CSF_IDX.FOR and MAKESAS.SAS.

F. REFERENCE

Kennedy, Eileen, James Ohls, Steven Carlson, Katherine Fleming. "The Healthy Eating Index: Final Report." Princeton, NJ: Mathematica Policy Research, Inc., December 1994.

APPENDIX A

INFORMATION ABOUT INPUT FILES

THE HEALTHY EATING INDEX CREATION PROCESS

INPUT FILES

CSF89.RAW

Source: U.S. Department of Agriculture, Human Nutrition Information Service
Contents: The 1989 Continuing Survey of Food Intake by Individuals (CSFII).   Complete documentation for this file is available from USDA.

CSF90.RAW

Source: U.S. Department of Agriculture, Human Nutrition Information Service
Contents: The 1990 Continuing Survey of Food Intake by Individuals (CSFII).   Complete documentation for this file is available from USDA.

POINTS.ASC

Source: Research conducted by Jim Ohls at Mathematica Policy Research, Inc.
Format: ASCII
Contents: A matrix which contains the maximum and minimum values for each Healthy Eating Index component, broken out by age and sex groups. The maximum value is equivalent to a score of 10 on the component; the minimum value is equivalent to a score of 0.

Columns	Variable	Notes
1-3	Sex	1=male; 2=female
4-7	Age groups	1=1-3; 2=4-6; 3=7-10; 4=11-14; 5=15-18; 6=19-24; 7=25-50; 8=51+
8-13	Index groups	1-10 correspond to the ten index components; 11 is total calories
14-22	Max point	
23-31	Min point	

SERVINGS.MPR

Source: Technical Assessment Systems (TAS), Inc.
Format: ASCII
Contents: Translates the 7-digit USDA food codes to food pyramid servings. The file is hierarchical. The first hierarchy is the USDA food code. The second hierarchy is the pyramid category. There is one second-level record for each pyramid category in each USDA food code.

Level 1:

Column	Variable	Notes
7-13	7 digit food code	
23-26	TAS code	Used only by TAS
38-39	# of pyramid groups	corresponds to number of second-level records

Level 2:

Column	Variable	Notes
12-13	Pyramid group code	1=dairy 2=protein 3=carbohydrate 4=vegetable 5=fruit 6=legume 9=sweeteners 10=fats&oils
17-26	# of grams/100g	Used for Group 9&10 serving sizes
30-39	# of servings/100g	Used for Group 1-6 serving sizes

VARSCORE.MPR

Source: TAS
Format: ASCII
Contents: Breaks down each each USDA 7 digit food code into its constituent codes. A hamburger, for example, could be broken down into a beef code and a bread code. This constituent code (CC) information is used to compute the variety component of the Healthy Eating Index.

Column	Variable	Notes
1-7	7 digit food code	
10	Mixture flag	"M" if the item is a mixture; blank otherwise
12-13	Not calculated flag	Indicates items whose components were not calculated
16-55	Label	Text accompanying each food code
56-62	7 digit food code	Used as constituent code if item is not a mixture
63-71	Dairy CC	
72-80	Meat CC1	first of three possible meat constituent codes
81-89	Meat CC2	second
90-98	Meat CC3	third
99-107	Grain CC1	
108-116	Grain CC2	
117-125	Vegetable CC1	
126-134	Vegetable CC2	
135-143	Fruit CC1	
144-152	Fruit CC2	
153-161	Legume CC1	
162-170	Legume CC2	

7DIGIT.CDE

Source: CSF89_2.DOC. This file is part of the CSFII documentation.
Format: ASCII
Contents: Contains the text labels that correspond to each 7 digit food code.

Column	Variable
1-7	7 digit USDA food code
8-69	Label

CSF_IDX.FOR

Input:	CSF89D4.BIN or CSF90D4.BIN, POINTS.ASC, MRG_TAS3.ASC
Output:	CSF_IDX4.BIN
Include files:	WORKCSF.INC, F7_CODE.INC, INDEX.INC
Language:	FORTRAN
Purpose:	Creates the Healthy Eating Index.
Calling Tree:	

```
(MAIN)
 |
 |——TIME [m]
 |
 |——DATE [m]
 |
 |——TIMER
 |   |
 |   └——3*GETTIM
 |       |
 |       └——TIME [m]
 |
 |——READ_AUXILLARY_DATA
 |
 |——2*READ_PERSON
 |
 |——CREATE_INDEX
 |   |
 |   └——BINARY_SEARCH_INT4
 |
 |——ZERO_INDEX
 |
 |——WRITE_PERSON
 |
 |——PRINT_PERSON
 |
 |——READ_PERSON
 |
 └——TIMER [d]
```
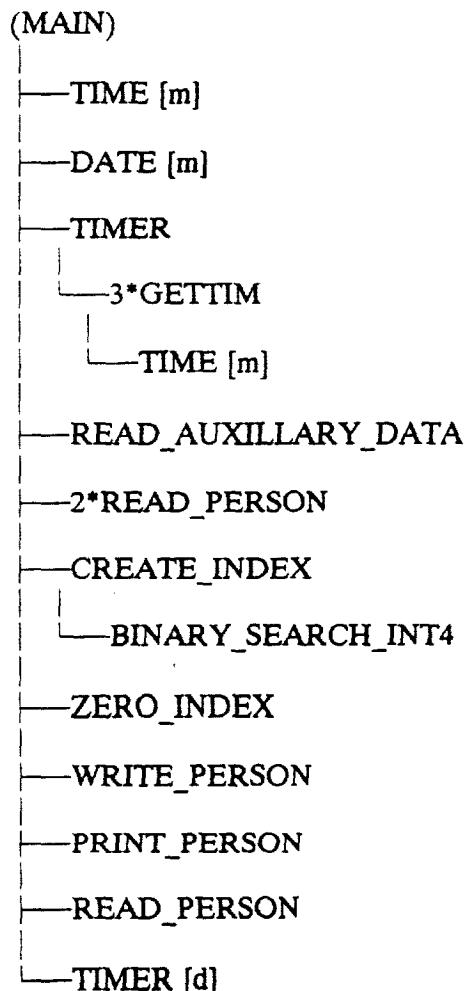
FIXDATA.FOR

Input:	CSF89.RAW
Output:	CSF89.FIX
Include files:	None
Language:	FORTRAN
Purpose:	Fixes a bad record on the 1989 CSFII. There is no equivalent program for the 1990 CSFII Healthy Eating Index.
Calling tree:	This program has no subroutines.

FIX_SERV.FOR

Input:	SERVINGS.MPR, 7DIGIT.CDE
Output:	FIX_SERV.ASC
Include files:	None
Language:	FORTRAN
Purpose:	Writes out a file (FIX_SERV.ASC) that combines the NCFS code and portion sizes from SERVINGS.MPR with the labels on 7DIGIT.CDE.
Calling tree:	This program has no subroutines.

MAKESAS.SAS

Input:	CSF_IDX4.BIN
Output:	CSFINDEX.SSD
Language:	SAS
Purpose:	Creates a SAS version of the Healthy Eating Index file.

MRG_TAS3.FOR

Input:	FIX_SERV.SRT, VARSCORE.MPR
Output:	MRG_TAS3.ASC
Include files:	None
Language:	FORTRAN
Purpose:	Writes out a file (MRG_TAS3.ASC) with the serving sizes, labels, and constituent component codes for every possible NCFS code.
Calling tree:	This program has no subroutines.

SORTSERV.BAT

Input:	FIX_SERV.ASC
Output:	FIX_SERV.SRT
Language:	DOS batch file that invokes OTSORT
Purpose:	Sorts FIX_SERV.ASC in ascending order by NCFS code.

XTCSF.FOR

Input: CSF89.FIX or CSF90.RAW
Output: CSF89D4.BIN or CSF90D4.BIN
Include files: XTCSF.INC, EXTVAR.INC
Language: FORTRAN
Purpose: Creates a person-level extract from the CSFII.
Calling tree:

```
(MAIN)
  |
  |——TIMER
  |   |
  |   └——3*GETTIM
  |       |
  |       └——TIME [m]
  |
  |——2*READ_HH
  |
  |——PRINT_HH
  |
  |——EXTRACT_DATA
  |
  |——READ_HH
  |
  └——TIMER [d]
```

13
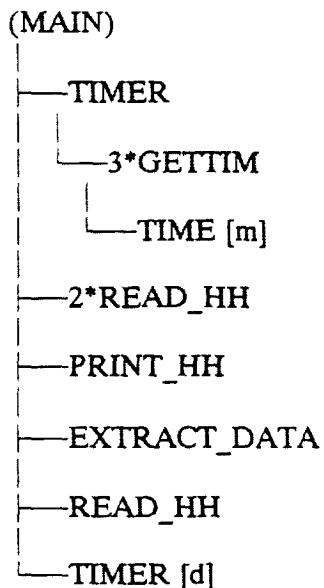
APPENDIX B

DESCRIPTION OF HEI ANALYSIS FILE

The SAS System

CONTENTS PROCEDURE

Data Set Name: OUT.CSFINDEX   Observations: 3997
Member Type: DATA   Variables: 176
Engine: V604   Indexes: 0
Created: 14:40 Friday, December 9, 1994   Observation Length: 1396
Last Modified: 14:40 Friday, December 9, 1994   Deleted Observations: 0
Protection: Compressed: NO
Data Set Type: Sorted: NO
Label:

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Label
57	ACTIVE_L	Num	8	444	Self-assesment of activity level
23	AGE	Num	8	172	Age in years
173	AGG_CNT	Num	8	1364	Used to calculate variety component
161	AVGLEGUM	Num	8	1268	Average daily legumes consumption
156	AVG_FG1	Num	8	1228	Average daily grains consumption
157	AVG_FG2	Num	8	1236	Average daily vegetables consumption
158	AVG_FG3	Num	8	1244	Average daily fruits consumption
159	AVG_FG4	Num	8	1252	Average daily dairy consumption
160	AVG_FG5	Num	8	1260	Average daily meat consumption
55	BMI	Num	8	428	Body mass index: wgt (kg)/hgt squared(m)
119	BMILK	Num	8	932	Human milk consumed? 1=yes 2=no
20	CHILDREN	Num	8	148	HH has members < age 20? 1=yes 2=no
162	COMP01	Num	8	1276	Grains HEI component
163	COMP02	Num	8	1284	Vegetables HEI component
164	COMP03	Num	8	1292	Fruit HEI component
165	COMP04	Num	8	1300	Dairy HEI component
166	COMP05	Num	8	1308	Meat HEI component
167	COMP06	Num	8	1316	All fat-pct HEI component
168	COMP07	Num	8	1324	Saturated fat pct HEI component
169	COMP08	Num	8	1332	Cholesterol HEI component
170	COMP09	Num	8	1340	Sodium HEI component
171	COMP10	Num	8	1348	Variety HEI component
64	D1_AMTUS	Num	8	500	Self-assesment of day 1 food/drink amt
59	D1_DATE	Num	8	460	Day 1 date of intake: day of month
61	D1_DAY	Num	8	476	Day 1 date of intake: day of week
63	D1_FEW	Num	8	492	Day 1 explanation for few/no food recs
58	D1_MNTH	Num	8	452	Day 1 date of intake: month
62	D1_NREC	Num	8	484	Day 1 number of food records
65	D1_RESUS	Num	8	508	Day 1 reason for difference
60	D1_YEAR	Num	8	468	Day 1 date of intake: year
72	D2_AMTUS	Num	8	564	Self-assesment of day 2 food/drink amt
67	D2_DATE	Num	8	524	Day 2 date of intake: day of month
69	D2_DAY	Num	8	540	Day 2 date of intake: day of week
71	D2_FEW	Num	8	556	Day 2 explanation for few/no food recs
66	D2_MNTH	Num	8	516	Day 2 date of intake: month
70	D2_NREC	Num	8	548	Day 2 number of food records
73	D2_RESUS	Num	8	572	Day 2 reason for difference
68	D2_YEAR	Num	8	532	Day 2 date of intake: year
80	D3_AMTUS	Num	8	628	Self-assesment of Day 3 food/drink amt
75	D3_DATE	Num	8	588	Day 3 date of intake: day of month
77	D3_DAY	Num	8	604	Day 3 date of intake: day of week
79	D3_FEW	Num	8	620	Day 3 explanation for few/no food recs
74	D3_MNTH	Num	8	580	Day 3 date of intake: month
78	D3_NREC	Num	8	612	Day 3 number of food records
81	D3_RESUS	Num	8	636	Day 3 reason for difference
76	D3_YEAR	Num	8	596	Day 3 date of intake: year
118	DAYCOD40	Num	8	924	Day of intake, type 40 record
19	DHKHH	Num	8	140	Did HH complete DHKS? 1=yes 2=no
37	DS_CALOR	Num	8	284	Source of calorie/weight loss diet
45	DS_DIAB	Num	8	348	Source of diabetic diet
39	DS_FAT	Num	8	300	Source of low fat/cholesterol diet
49	DS_HFIB	Num	8	380	Source of high fiber diet
47	DS_LFIB	Num	8	364	Source of low fiber diet
51	DS_OTHER	Num	8	396	Source of other diet
41	DS_SALT	Num	8	316	Source of low salt diet
43	DS_SUGAR	Num	8	332	Source of low sugar/sugar free diet
36	DT_CALOR	Num	8	276	On calorie/weight loss diet? 1=yes 2=no
44	DT_DIAB	Num	8	340	On a diabetic diet? 1=yes 2=no
38	DT_FAT	Num	8	292	On low fat/cholesterol diet? 1=yes 2=no

17

48	DT_HFIB	Num	8	372	On a high fiber diet? 1=yes 2=no
46	DT_LFIB	Num	8	356	On a low fiber diet? 1=yes 2=no
50	DT_OTHER	Num	8	388	On another diet? 1=yes 2=no
40	DT_SALT	Num	8	308	On a low salt diet? 1=yes 2=no
42	DT_SUGAR	Num	8	324	On low sugar/sugar free diet? 1=yes 2=no
89	F1A	Num	8	696	Diet awareness question: fruit servings
90	F1B	Num	8	704	Diet awareness question: vegetable svgs
91	F1C	Num	8	712	Diet awareness question: dairy servings
92	F1D	Num	8	720	Diet awareness question: grain servings
93	F1E	Num	8	728	Diet awareness question: meat servings
94	F4A	Num	8	736	Important to: avoid salt?
95	F4B	Num	8	744	Important to: avoid saturated fat?
96	F4C1	Num	8	752	Important to: eat fruits & vegs?
97	F4C2	Num	8	760	Important to: eat plenty of vegs?
98	F4C3	Num	8	768	Important to: eat plenty of fruits?
99	F4D	Num	8	776	Important to: avoid sugar?
100	F4E1	Num	8	784	Important to: drink in moderation?
101	F4E2	Num	8	792	Important to: limit alcohol?
102	F4F	Num	8	800	Important to: eat adequate fiber?
103	F4G	Num	8	808	Important to: eat adequate starch?
104	F4H	Num	8	816	Important to: eat variety of foods?
105	F4I	Num	8	824	Important to: maintain a healthy weight?
106	F4J	Num	8	832	Important to: avoid fat?
107	F4K	Num	8	840	Important to: avoid cholesterol?
108	F4L	Num	8	848	Important to eat 6+ servings of grain?
109	F4M	Num	8	856	Important to: eat plenty of grain?
110	F4N	Num	8	864	Important to: eat calcium-rich foods?
111	F4O	Num	8	872	Important to: eat 2+ dairy servings?
174	FATRATIO	Num	8	1372	Fat as % of total calories
150	FG1	Num	8	1180	Total grains consumed over 3 days
151	FG2	Num	8	1188	Total vegetables consumed over 3 days
152	FG3	Num	8	1196	Total fruits consumed over 3 days
153	FG4	Num	8	1204	Total dairy consumed over 3 days
154	FG5	Num	8	1212	Total meat consumed over 3 days
155	FG6	Num	8	1220	Total legumes consumed over 3 days
6	FOODDESC	Num	8	40	Measure of food security
13	GEOG	Num	8	92	Geographic division (9 possible regions)
176	HD_INDEX	Num	8	1388	Healthy eating index value
21	HEADSTAT	Num	8	156	Head of household status
4	HHID_15	Num	8	24	Household ID number on type 15 record
11	HHID_20	Num	8	76	Household ID number on type 20 record
115	HHID_40	Num	8	900	Household ID number on type 40 record
86	HHID_50	Num	8	672	Household ID number on type 50 record
15	HHSZ	Num	8	108	HH size excluding roomers
34	HLTHDIET	Num	8	260	Healthfulness of diet
18	H_RCVFS	Num	8	132	Is HH receiving food stamps? 1=yes 2=no
16	INCOME	Num	8	116	HH income for the previous year
56	LEISURE	Num	8	436	Physical activity during leisure
5	LGRADE15	Num	8	32	Highest grade finished by either HH head
28	NUMDAYS	Num	8	212	# of days of complete intake records
27	ORIGIN	Num	8	204	Hispanic origin: 1=Hisp 2=Non-Hisp
17	PCTPOV	Num	8	124	Income as % of poverty threshold
25	PL_STAT	Num	8	188	Pregnant/lactating status
26	RACE_20	Num	8	196	Race: 1=white 2=black 3=Asian 4=Aleut
117	RACE_40	Num	8	916	Race on type 40 record
88	RACE_50	Num	8	688	Race on type 50 record
120	RDA1	Num	8	940	% consumed of food energy RDA
121	RDA2	Num	8	948	% consumed of protein RDA
122	RDA3	Num	8	956	% consumed of Vitamin A (IU) RDA
123	RDA4	Num	8	964	% consumed of Vitamin A (RE) RDA
124	RDA5	Num	8	972	% consumed of Vitamin E (Alpha-TE) RDA
125	RDA6	Num	8	980	% consumed of Vitamin C RDA
126	RDA7	Num	8	988	% consumed of Thiamin RDA
127	RDA8	Num	8	996	% consumed of Riboflavin RDA
128	RDA9	Num	8	1004	% consumed of Niacin RDA
129	RDA10	Num	8	1012	% consumed of Vitamin B-6 RDA
130	RDA11	Num	8	1020	% consumed of Folate RDA
131	RDA12	Num	8	1028	% consumed of Vitamin B-12 RDA
132	RDA13	Num	8	1036	% consumed of Calcium RDA
133	RDA14	Num	8	1044	% consumed of Phosphorus RDA
134	RDA15	Num	8	1052	% consumed of Magnesium RDA
135	RDA16	Num	8	1060	% consumed of Iron RDA
136	RDA17	Num	8	1068	% consumed of Zinc RDA
1	RT_15	Char	4	4	Record type on type 15 record
8	RT_20	Char	4	56	Record type on type 20 record
112	RT_40	Char	4	880	Record type on type 40 record

83	RT_50	Char	4	652	Record type on type 50 record
32	R_CNTL	Num	8	244	# of complete school lunches gotten/week
30	R_EMP	Num	8	228	Employment status
33	R_FREEL	Num	8	252	Lunches free, reduced, or full price?
54	R_HGT	Num	8	420	Height
31	R_LGRADE	Num	8	236	Highest grade completed for persons > 15
12	R_NUM_20	Num	8	84	Type 20 person number
116	R_NUM_40	Num	8	908	Person number on type 40 record
87	R_NUM_50	Num	8	680	Person number on type 50 record
22	R_RCVFS	Num	8	164	Is person receiving food stamps? 1=y 2=n
53	R_WGT	Num	8	412	Weight
29	R_WIC	Num	8	220	Is person receiving WIC? 1=yes 2=no
3	SAMPLE15	Num	8	16	Type 15 sample: 1 = Basic 2 = Low Income
10	SAMPLE20	Num	8	68	Type 20 sample: 1 = Basic 2 = Low Income
114	SAMPLE40	Num	8	892	Type 40 sample: 1 = Basic 2 = Low Income
85	SAMPLE50	Num	8	664	Type 50 sample: 1 = Basic 2 = Low Income
175	SAT_FAT	Num	8	1380	Saturated fat as % of total calories
24	SEX	Num	8	180	Sex: 1 = male 2 = female
35	SPECDIET	Num	8	268	On a special diet? 1=yes 2=no
172	SPLITCNT	Num	8	1356	Used to calculate variety component
7	SURPLUS	Num	8	48	Did HH receive USDA surplus? 1=yes 2=no.
137	TOTNUT1	Num	8	1076	Total grams of water consumed
138	TOTNUT2	Num	8	1084	Total kilocalories of food energy
139	TOTNUT3	Num	8	1092	Total grams of protein
140	TOTNUT4	Num	8	1100	Total grams of fat
141	TOTNUT5	Num	8	1108	Total grams of saturated fatty acids
142	TOTNUT6	Num	8	1116	Total g of monounsaturated fatty acids
143	TOTNUT7	Num	8	1124	Total g of polyunsaturated fatty acids
144	TOTNUT8	Num	8	1132	Total milligrams of cholesterol
145	TOTNUT9	Num	8	1140	Total grams of carbohydrates
146	TOTNUT10	Num	8	1148	Total grams of dietary fiber
147	TOTNUT11	Num	8	1156	Total grams of alcohol
148	TOTNUT29	Num	8	1164	Total milligrams of sodium
149	TYP30CNT	Num	8	1172	# of type 30 (food item) records
14	URB	Num	8	100	Urbanization: 1=city 2=suburb 3=nonmetro
52	VEGET	Num	8	404	Do you consider yourself a vegetarian?
82	WGT_3DAY	Num	8	644	Weight for persons w/3 days of intake
2	YEAR_15	Num	8	8	Year of survey on type 15 record
9	YEAR_20	Num	8	60	Year of survey on type 20 record
113	YEAR_40	Num	8	884	Year of survey on type 40 record
84	YEAR_50	Num	8	656	Year of survey on type 50 record

APPENDIX C

FORTRAN PROGRAM WHICH CREATES THE INDEX VARIABLES

```
C*****************************************************
C--- SAVED AS CSF_IDX.FOR
C--- PURPOSE:  READ A BINARY EXTRACT OF THE CSFII
C---           DATA AND CREATE A HEALTHY DIET
C---           INDEX.
C---           THIS VERSION prompts for the day.
C---
C--- PREV PGM: XTCSF.FOR
C--- PREV PGM: MRG_TAS2.FOR
C--- PREV PGM: FIX_SERV.FOR
C--- PREV PGM: SORTSERV.BAT
C*****************************************************

      IMPLICIT NONE

      INCLUDE 'WORKCSF.INC'
      INCLUDE 'F7_CODE.INC'
      INCLUDE 'INDEX.INC'

      REAL*8 AVG_COMP01
      REAL*8 AVG_COMP02
      REAL*8 AVG_COMP03
      REAL*8 AVG_COMP04
      REAL*8 AVG_COMP05
      REAL*8 AVG_COMP06
      REAL*8 AVG_COMP07
      REAL*8 AVG_COMP08
      REAL*8 AVG_COMP09
      REAL*8 AVG_COMP10

      CHARACTER*12 INFILE_NAME
      CHARACTER*12 OUTFILE_NAME
      CHARACTER*12 PRFILE_NAME
      CHARACTER*12 C_TIME
      CHARACTER*12 C_DATE
C*************************************************
C------- OPEN FILES               ---*
C*************************************************

      PRFILE = 8
      KEOF = 1

100   CONTINUE
C---- (WHICH_DAY: 1=DAY1, 2=DAY2, 3=DAY3, 4=ALL 3 DAYS)
      WRITE(6,*) 'WHICH FILE DO YOU WANT?'
      WRITE(6,*) 'PLEASE ENTER:'
      WRITE(6,*) '  1 = DAY 1 FILE'
      WRITE(6,*) '  4 = 3 DAY FILE'
      WRITE(6,*) '  0 = STOP RUN'
      WRITE(6,*) 'PLEASE ENTER: '
      READ(5,*) WHICH_DAY
      WRITE(6,*) ' USING: ',WHICH_DAY
      WRITE(6,*) ' '

      TOT_COMP01 = 0.0
      TOT_COMP02 = 0.0
      TOT_COMP03 = 0.0
      TOT_COMP04 = 0.0
      TOT_COMP05 = 0.0
      TOT_COMP06 = 0.0
      TOT_COMP07 = 0.0
      TOT_COMP08 = 0.0
      TOT_COMP09 = 0.0
      TOT_COMP10 = 0.0
      TOT_WGT = 0.0

      DAY1 = .FALSE.
      DAY2 = .FALSE.
      DAY3 = .FALSE.
      IF(WHICH_DAY .EQ. 4) THEN ! ALL THREE DAYS
         INFILE_NAME = 'CSF90D4.BIN'
         OUTFILE_NAME = 'CSF_IDX4.BIN'
         PRFILE_NAME = 'CSF_IDX4.OUT'
         DAY1 = .TRUE.
         DAY2 = .TRUE.
         DAY3 = .TRUE.
```

23

```fortran
          ELSE IF(WHICH_DAY .EQ. 1) THEN ! DAY 1
             INFILE_NAME = 'CSF90D1.BIN'
             OUTFILE_NAME = 'CSF_IDX1.BIN'
             PRFILE_NAME = 'CSF_IDX1.OUT'
             DAY1 = .TRUE.
          ELSE IF(WHICH_DAY .EQ. 0) THEN ! STOP
             WRITE(6,*) ' STOPPING RUN'
             STOP
          ELSE
             WRITE(6,*) ' SORRY; YOU HAVE ENTERED AN INVALID CODE.'
             WRITE(6,*) ' PLEASE RE-ENTER'
             GO TO 100
          END IF

          OPEN(UNIT = 8,FILE=PRFILE_NAME)
          OPEN(UNIT = 9
     &       ,FILE=INFILE_NAME
     &       ,ACCESS='TRANSPARENT'
     &       ,BLOCKSIZE=32000)
          OPEN(UNIT = 10
     &       ,FILE=OUTFILE_NAME
     &       ,ACCESS='TRANSPARENT'
     &       ,BLOCKSIZE=32000)


          CALL TIME(C_TIME)
          CALL DATE(C_DATE)
          WRITE(PRFILE,*) C_DATE
          WRITE(PRFILE,*) C_TIME
          WRITE(PRFILE,*) 'OUTPUT FILE = ',OUTFILE_NAME
          WRITE(PRFILE,*) 'PRINT FILE  = ',PRFILE_NAME
          WRITE(PRFILE,*) 'WHICH DAY   = ',WHICH_DAY
          CALL TIMER(KEOF,PRFILE)
          CALL READ_AUXILLARY_DATA
          CALL READ_PERSON

          KEOF = 2
          CALL READ_PERSON
          KFREQ = 0
C***********************************
C------ MAIN PROCESSING LOOP    ----*
C***********************************
200   CONTINUE


          IF(TYPE_20_CNT .LE. 1) KFREQ = 1
C****  IF(TYPE_20_CNT .GT. 496 .AND. TYPE_20_CNT .LE. 500) KFREQ = 1
C****  IF(AGE .LE. 10) WRITE(6,*) ' AGE = ',AGE,' FOR: ', TYPE_20_CNT
C****  IF(HHID_15 .EQ. 10022 .OR. HHID_15 .EQ. 10025) KFREQ = 1
C****  IF(HHID_15 .EQ. 10027 .OR. HHID_15 .EQ. 10041) KFREQ = 1
          IF(HHID_15 .EQ. 10009 .OR. HHID_15 .EQ. 10010) THEN
             KFREQ = 9 | 0 VARIETY
             WRITE(8,*) ' --- FOUND A ZERO VARIETY RECORD ----'
          END IF
C****  IF(HHID_15 .EQ. 10012 .OR. HHID_15 .EQ. 10234) THEN
C****     KFREQ = 9 | 0 VARIETY
C****     WRITE(8,*) ' --- FOUND A ZERO VARIETY RECORD ----'
C****  END IF
C     IF(HHID_15 .EQ. 21159 .OR. HHID_15 .EQ. 26092) THEN
C        KFREQ = 9 | 0 VARIETY
C        WRITE(8,*) ' --- FOUND A ZERO VARIETY RECORD ----'
C     END IF
          IF(HHID_15 .EQ. 10326
     &       .OR. HHID_15 .EQ. 12242
     &       .OR. HHID_15 .EQ. 20125
     &          ) THEN
             KFREQ = 1 |
             WRITE(8,*) ' --- FOUND A JIM DEBUG CASE ----'
          END IF

          IF(MOD(TYPE_20_CNT,100) .EQ. 0) THEN
            WRITE(6,2020) TYPE_20_CNT
2020        FORMAT('+TYPE_20_CNT = ',I6)
          END IF
```

24

```fortran
      IF(TYPE_30_CNT .GT. 0) THEN
         CALL CREATE_INDEX
      ELSE
         CALL ZERO_INDEX
      END IF

      IF(.NOT. HUMAN_MILK .AND. AGE .GT. 1) CALL WRITE_PERSON

      IF(KFREQ .GT. 0) CALL PRINT_PERSON

      KFREQ = 0
      CALL READ_PERSON
      IF(KEOF .EQ. 3) GO TO 900

      GO TO 200  ! READ NEXT PERSON

900   CONTINUE
C***********************************
C       EOF
C***********************************

      WRITE(PRFILE,9010) HUMAN_MILK_CNT
     &                   ,OUT_CNT
9010  FORMAT('  HUMAN_MILK_CNT = ',I6
     &     /,'  OUT_CNT        = ',I6
     &        )

      AVG_COMP01 = TOT_COMP01 / TOT_WGT
      AVG_COMP02 = TOT_COMP02 / TOT_WGT
      AVG_COMP03 = TOT_COMP03 / TOT_WGT
      AVG_COMP04 = TOT_COMP04 / TOT_WGT
      AVG_COMP05 = TOT_COMP05 / TOT_WGT
      AVG_COMP06 = TOT_COMP06 / TOT_WGT
      AVG_COMP07 = TOT_COMP07 / TOT_WGT
      AVG_COMP08 = TOT_COMP08 / TOT_WGT
      AVG_COMP09 = TOT_COMP09 / TOT_WGT
      AVG_COMP10 = TOT_COMP10 / TOT_WGT

      WRITE(PRFILE,9020) TOT_WGT
     &                   ,AVG_COMP01
     &                   ,AVG_COMP02
     &                   ,AVG_COMP03
     &                   ,AVG_COMP04
     &                   ,AVG_COMP05
     &                   ,AVG_COMP06
     &                   ,AVG_COMP07
     &                   ,AVG_COMP08
     &                   ,AVG_COMP09
     &                   ,AVG_COMP10
9020  FORMAT(///
     &     /,'  TOT_WGT    = ',F12.2
     &     /,'  AVG_COMP01 = ',F12.2
     &     /,'  AVG_COMP02 = ',F12.2
     &     /,'  AVG_COMP03 = ',F12.2
     &     /,'  AVG_COMP04 = ',F12.2
     &     /,'  AVG_COMP05 = ',F12.2
     &     /,'  AVG_COMP06 = ',F12.2
     &     /,'  AVG_COMP07 = ',F12.2
     &     /,'  AVG_COMP08 = ',F12.2
     &     /,'  AVG_COMP09 = ',F12.2
     &     /,'  AVG_COMP10 = ',F12.2
     &        )
      CALL TIMER(KEOF,PRFILE)
      STOP
      END


      SUBROUTINE READ_AUXILLARY_DATA
C***************************************************
C--- READ IN TAS 7-DIGIT FOOD CODE FILE ALONG
C--- WITH THE SERVINGS PORTIONS AND LABEL IN 1 FILE
C--- READ IN POINTS ARRAY FROM THE SECOND FILE
C***************************************************

      IMPLICIT NONE
      INTEGER*4 I
```

25

```fortran
      INTEGER*4 J
      INTEGER*4 ISEX
      INTEGER*4 IAGE
      INTEGER*4 IGRP
      CHARACTER*1 FF/Z'0C'/

      INCLUDE 'WORKCSF.INC'
      INCLUDE 'F7_CODE.INC'
C     CHARACTER*1 SEX_GRP(2) /'M','F'/
      CHARACTER*5 AGE_GRP(8)/
     &    ' 1-3'
     & , ' 4-6'
     & , ' 7-10'
     & ,'11-14'
     & ,'15-18'
     & ,'19-24'
     & ,'25-50'
     & ,' 51+'
     & /
      CHARACTER*9 FOOD_GRP(11)/
     &    'Grain'
     & ,'Vegetable'
     & ,'Fruit'
     & ,'Dairy'
     & ,'Meat'
     & ,'All Fat'
     & ,'Sat Fat'
     & ,'Cholest'
     & ,'Sodium'
     & ,'Variety'
     & ,'Calories'
     & /

      WRITE(PRFILE,1000)
1000  FORMAT(' ---- IN READ_AUXILLARY_DATA ----')


      OPEN(UNIT = 1,FILE='MRG_TAS3.ASC')
      OPEN(UNIT = 2,FILE='POINTS.ASC')

      MAX_CODES = 0
      I = 1

      WRITE(6,*) ' READING THE MERGED TAS-FOODLABEL FILE....'

100   CONTINUE
C----- read the 7 digit foodcodes, labels, and servings per 100 grams
      READ(1,1010,END=200) FOODCODE7(I)
     &                     ,(SERVINGS(I,J),J=1,10)
     &                     ,LABEL7(I)
     &                     ,(VFC(I,J),J=1,13)
1010  FORMAT(I7,10F7.2,A40,13I8)

      IF(I .LE. 20) THEN
          WRITE(PRFILE,1200)
     &                      FOODCODE7(I)
     &                      ,LABEL7(I)
     &                      ,(SERVINGS(I,J),J=1,10)
     &                      ,(VFC(I,J),J=1,13)
1200      FORMAT(' '
     &            ,' FOODCODE: ',I7
     &            ,' LABEL: ',A40
     &          /,' SERVINGS =',10F8.2
     &          /,' VFC      =',13I8
     &            )
      END IF
      I = I + 1
      GO TO 100


C----- read in the points file
200   CONTINUE
      MAX_CODES = I ! used in binary search for food codes
      WRITE(PRFILE,2010) FF,MAX_CODES
2010  FORMAT(A1,//,' ** At End of foodcode file:'
```

26

```
     &     ,/,' Total number of records read in: ', I8)


      WRITE(6,*) ' READING THE POINTS.ASC FILE....'
      WRITE(6,*) ' '

      READ(2,2020) ! SKIP THE HEADER LINE AT THE TOP
2020  FORMAT(A50)

300   CONTINUE
      READ(2,3010,END=900) ISEX
     &                     ,IAGE
     &                     ,IGRP
     &                     ,MAX_POINT(ISEX,IAGE,IGRP)
     &                     ,MIN_POINT(ISEX,IAGE,IGRP)
3010  FORMAT(T3,I1,T7,I1,T12,I2,T15,F8.3,T24,F8.3,T35,A50)

C----- convert variety from a 3 day level to a one day level
      IF(IGRP .EQ. 10 .AND. WHICH_DAY .NE. 4) THEN
         MAX_POINT(ISEX,IAGE,IGRP) = MAX_POINT(ISEX,IAGE,IGRP) / 2.0
         MIN_POINT(ISEX,IAGE,IGRP) = MIN_POINT(ISEX,IAGE,IGRP) / 2.0
      END IF
C----- CONVERT MEAT FROM OUNCES TO SERVINGS
      IF(IGRP .EQ. 5) THEN ! convert meats from oz to servings
         MAX_POINT(ISEX,IAGE,IGRP) = MAX_POINT(ISEX,IAGE,IGRP) / 2.5
         MIN_POINT(ISEX,IAGE,IGRP) = MIN_POINT(ISEX,IAGE,IGRP) / 2.5
      END IF
C      WRITE(PRFILE,3020) SEX_GRP(ISEX)
C     &                   ,AGE_GRP(IAGE)
C     &                   ,FOOD_GRP(IGRP)
C     &                   ,MAX_POINT(ISEX,IAGE,IGRP)
C     &                   ,MIN_POINT(ISEX,IAGE,IGRP)
C3020  FORMAT(
C     &        '  ',A1
C     &      ,'  ',A5
C     &      ,'  ',A9
C     &      ,'  MAX_POINT =',F8.2
C     &      ,'  MIN_POINT =',F8.2
C     &        )

      GO TO 300

900   CONTINUE
C----------------------------
C--- eof for both files
C----------------------------
      WRITE(PRFILE,9010) FF
9010  FORMAT(A1,T45,'MALES')
      WRITE(PRFILE,9020) (AGE_GRP(I),I=1,8)
9020  FORMAT(/,'        Age Group',T18,8A10)
      DO IGRP = 1,11 ! FOOD GROUPS
         WRITE(PRFILE,9050) FOOD_GRP(IGRP)
     &                     ,(MAX_POINT(1,IAGE,IGRP),IAGE=1,8)
9050     FORMAT(/,' ',A10,' MAX: ',8F10.3)
         WRITE(PRFILE,9060) FOOD_GRP(IGRP)
     &                     ,(MIN_POINT(1,IAGE,IGRP),IAGE=1,8)
9060     FORMAT(' ',A10,' MIN: ',8F10.3)
      END DO

      WRITE(PRFILE,9110)
9110  FORMAT(///,T44,'FEMALES')
      WRITE(PRFILE,9020) (AGE_GRP(I),I=1,8)
      DO IGRP = 1,11 ! FOOD GROUPS
         WRITE(PRFILE,9050) FOOD_GRP(IGRP)
     &                     ,(MAX_POINT(2,IAGE,IGRP),IAGE=1,8)
         WRITE(PRFILE,9060) FOOD_GRP(IGRP)
     &                     ,(MIN_POINT(2,IAGE,IGRP),IAGE=1,8)
      END DO
      WRITE(PRFILE,9910) FF
9910  FORMAT(
     &       /,' ---- END OF READ_AUXILLARY_DATA ----',A1
     &       )

      CLOSE(1)
      CLOSE(2)
```

27

```
      RETURN
      END

      SUBROUTINE READ_PERSON
C****************************************************
C--- READ IN AN ENTIRE HOUSEHOLD AT A TIME
C****************************************************

      IMPLICIT NONE
      INTEGER*4 J

      CHARACTER*1 FF /Z'0C'/
                                                                    WORKCSF
      INCLUDE 'WORKCSF.INC'

C      IF(KFREQ .GT. 0) '---- IN READ_PERSON ---'
       IF(KFREQ .GT. 0) THEN
          WRITE(PRFILE,1100) FF,IN_CNT
1100      FORMAT(A1,'---- IN READ_PERSON ---'
     &            ,' IN_CNT = ',I8)
       END IF

       IF(KEOF .EQ. 2) THEN
          GO TO 200
       ELSE IF(KEOF .EQ. 1) THEN
          GO TO 100
       ELSE
          GO TO 900
       END IF

100   CONTINUE
       IN_CNT = 0
       TYPE_20_CNT = 0
       TYPE_30_CNT = 0

       RETURN


200   CONTINUE
C***********************************
C------ READ A PERSON RECORD (TYPE 20)
C------ AND THEN ALL TYPE 30'S FOR
C------ THAT PERSON
C***********************************

          READ(9,END=900)
C--- TYPE 15
     &          RT_15      ! TYPE 15 VARIABLE
     &         ,YEAR_15    ! TYPE 15 VARIABLE
     &         ,SAMPLE_15  ! TYPE 15 VARIABLE
     &         ,HHID_15    ! TYPE 15 VARIABLE
     &         ,M_LGRADE   ! TYPE 15 VARIABLE
     &         ,F_LGRADE   ! TYPE 15 VARIABLE
     &         ,FOODSEC    ! TYPE 15 VARIABLE
     &         ,SURPLUS    ! TYPE 15 VARIABLE
C--- TYPE 20
     &         ,RT_20      ! TYPE 15-20 VARIABLE
     &         ,YEAR_20    ! TYPE 15-20 VARIABLE
     &         ,SAMPLE_20  ! TYPE 15-20 VARIABLE
     &         ,HHID_20    ! TYPE 15-20 VARIABLE
     &         ,R_NUM_20   ! TYPE 20 VARIABLE
     &         ,GEOG       ! TYPE 20 VARIABLE
     &         ,URB        ! TYPE 20 VARIABLE
     &         ,HHSZ       ! TYPE 20 VARIABLE
     &         ,INCOME     ! TYPE 20 VARIABLE
     &         ,PCTPOV     ! TYPE 20 VARIABLE
     &         ,H_RCVFS    ! TYPE 20 VARIABLE
     &         ,DHKHH      ! TYPE 20 VARIABLE
     &         ,CHILDREN   ! TYPE 20 VARIABLE
     &         ,HEADSTAT   ! TYPE 20 VARIABLE
     &         ,R_RCVFS    ! TYPE 20 VARIABLE
     &         ,AGE        ! TYPE 20 VARIABLE
     &         ,SEX        ! TYPE 20 VARIABLE
     &         ,PL_STAT    ! TYPE 20 VARIABLE
     &         ,RACE_20    ! TYPE 20 VARIABLE
     &         ,ORIGIN     ! TYPE 20 VARIABLE
```
23

```
&          ,NUMDAYS   ! TYPE 20 VARIABLE
&          ,R_WIC     ! TYPE 20 VARIABLE
&          ,R_EMP     ! TYPE 20 VARIABLE
&          ,R_LGRADE  ! TYPE 20 VARIABLE
&          ,R_CNTL    ! TYPE 20 VARIABLE
&          ,R_FREEL   ! TYPE 20 VARIABLE
&          ,HLTHDIET  ! TYPE 20 VARIABLE
&          ,SPECDIET  ! TYPE 20 VARIABLE
&          ,DT_CALOR  ! TYPE 20 VARIABLE  !    136      1   Are you on a calorie/weight loss diet?
       1 - Yes
&          ,DS_CALOR  ! TYPE 20 VARIABLE  !    137      1   Source of calorie/weight loss diet.
       1 - Doctor, dietician
&          ,DT_FAT    ! TYPE 20 VARIABLE  !    138      1   Are you on a low fat/cholesterol diet?
       1 - Yes
&          ,DS_FAT    ! TYPE 20 VARIABLE  !    139      1   Source of low fat/cholesterol diet.
       1 - Doctor, dietician
&          ,DT_SALT   ! TYPE 20 VARIABLE  !    140      1   Are you on a low salt diet?
       1 - Yes
&          ,DS_SALT   ! TYPE 20 VARIABLE  !    141      1   Source of low salt diet.
       1 - Doctor, dietician
&          ,DT_SUGAR  ! TYPE 20 VARIABLE  !    142      1   Are you on a low sugar/sugar free diet?
       1 - Yes
&          ,DS_SUGAR  ! TYPE 20 VARIABLE  !    143      1   Source of low sugar/sugar free diet.
       1 - Doctor, dietician
&          ,DT_DIAB   ! TYPE 20 VARIABLE  !    144      1   Are you on a diabetic diet?
       1 - Yes
&          ,DS_DIAB   ! TYPE 20 VARIABLE  !    145      1   Source of diabetic diet.
       1 - Doctor, dietician
&          ,DT_LFIB   ! TYPE 20 VARIABLE  !    146      1   Are you on a low fiber diet?
       1 - Yes
&          ,DS_LFIB   ! TYPE 20 VARIABLE  !    147      1   Source of low fiber diet.
       1 - Doctor, dietician
&          ,DT_HFIB   ! TYPE 20 VARIABLE  !    148      1   Are you on a high fiber diet?
       1 - Yes
&          ,DS_HFIB   ! TYPE 20 VARIABLE  !    149      1   Source of high fiber diet.
       1 - Doctor, dietician
&          ,DT_OTHER  ! TYPE 20 VARIABLE  !    150      1   Are you on some other type of diet?
       1 - Yes
&          ,DS_OTHER  ! TYPE 20 VARIABLE  !    151      1   Source of other diet.
       1 - Doctor, dietician
&          ,VEGET     ! TYPE 20 VARIABLE  !    152      1   Do you consider yourself to be a vegetarian?
       1 - Yes
&          ,R_WGT     ! TYPE 20 VARIABLE
&          ,R_HGT     ! TYPE 20 VARIABLE
&          ,BMI       ! TYPE 20 VARIABLE
&          ,LEISURE   ! TYPE 20 VARIABLE  !    207      1   What would you say your usual level of
       1 - Heavy / rigorous
&          ,ACTIVE_L  ! TYPE 20 VARIABLE  !    208      1   Compared with most people your age and sex,
       1 - More active
&          ,D1_MNTH   ! TYPE 20 VARIABLE  ! 265-266    2   Date of intake: month.
       1 - January
&          ,D1_DATE   ! TYPE 20 VARIABLE  ! 267-268    2   Date of intake: day of the month.
       1 - 31
&          ,D1_YEAR   ! TYPE 20 VARIABLE  !    269      1   Date of intake: year.
       9 - 1989
&          ,D1_DAY    ! TYPE 20 VARIABLE  !    270      1   Date of intake: day of the week.
       1 - Sunday
&          ,D1_NREC   ! TYPE 20 VARIABLE
&          ,D1_FEW    ! TYPE 20 VARIABLE  !    273      1   Explanation for few or no food records.
       3 - Ill, no food or beverage
&          ,D1_AMTUS  ! TYPE 20 VARIABLE  !    281      1   Would you say that the amount of food and drink
       1 - Less than usual
&          ,D1_RESUS  ! TYPE 20 VARIABLE  !    282      1   What best describes the reason for the
       1 - Sick or ill
&          ,D2_MNTH   ! TYPE 20 VARIABLE  ! 285-286    2   Date of intake: month.
       1 - January
&          ,D2_DATE   ! TYPE 20 VARIABLE  ! 287-288    2   Date of intake: day of the month.
       1 - 31
&          ,D2_YEAR   ! TYPE 20 VARIABLE  !    289      1   Date of intake: year.
       9 - 1989
&          ,D2_DAY    ! TYPE 20 VARIABLE  !    290      1   Date of intake: day of the week.
       1 - Sunday
&          ,D2_NREC   ! TYPE 20 VARIABLE
&          ,D2_FEW    ! TYPE 20 VARIABLE  !    293      1   Explanation for few or no food records.
       3 - Ill, no food or beverage
```

```
&        ,D2_AMTUS  ! TYPE 20 VARIABLE  !    301      1   Would you say that the amount of food and drink
         1 - Less than usual
&        ,D2_RESUS  ! TYPE 20 VARIABLE  !    302      1   What best describes the reason for the
         1 - Sick or ill
&        ,D3_MNTH   ! TYPE 20 VARIABLE  ! 306-307    2   Date of intake: month.
         1 - January
&        ,D3_DATE   ! TYPE 20 VARIABLE  ! 308-309    2   Date of intake: day of the month.
         1 - 31
&        ,D3_YEAR   ! TYPE 20 VARIABLE  !    310      1   Date of intake: year.
         9 - 1989
&        ,D3_DAY    ! TYPE 20 VARIABLE  !    311      1   Date of intake: day of the week.
         1 - Sunday
&        ,D3_NREC   ! TYPE 20 VARIABLE
&        ,D3_FEW    ! TYPE 20 VARIABLE  !    314      1   Explanation for few or no food records.
         3 - Ill, no food or beverage
&        ,D3_AMTUS  ! TYPE 20 VARIABLE  !    322      1   Would you say that the amount of food and drink
         1 - Less than usual
&        ,D3_RESUS  ! TYPE 20 VARIABLE  !    323      1   What best describes the reason for the
         1 - Sick or ill
&        ,WGT_3DAY  ! TYPE 20 VARIABLE  ! 578-580    3
C--- TYPE 50
&        ,RT_50     ! TYPE 50 VARIABLE  !    1-2      2   Record type.
         50 - All records
&        ,YEAR_50   ! TYPE 50 VARIABLE  !     3       1   Year of survey.
         1 - 1989
&        ,SAMPLE_50 ! TYPE 50 VARIABLE  !     4       1   Sample.
         1 - Basic
&        ,HHID_50   ! TYPE 50 VARIABLE  !    5-9      5   Household identification number.
&        ,R_NUM_50  ! TYPE 50 VARIABLE  !   10-11     2   Line number of DHKS respondent.
&        ,RACE_50   ! TYPE 50 VARIABLE  !    66       1   Race.
         1 - White
&        ,F1A       ! TYPE 50 VARIABLE  ! 153-154    2   1989:   Blank, not asked.
         0 - 97
&        ,F1B       ! TYPE 50 VARIABLE  ! 155-156    2   1989:   Blank, not asked.
         0 - 97
&        ,F1C       ! TYPE 50 VARIABLE  ! 157-158    2   1989:   Blank, not asked.
         0 - 97
&        ,F1D       ! TYPE 50 VARIABLE  ! 159-160    2   1989:   Blank, not asked.
         0 - 97
&        ,F1E       ! TYPE 50 VARIABLE  ! 161-162    2   1989:   Blank, not asked.
         0 - 97
&        ,F4A       ! TYPE 50 VARIABLE  !    195      1   1989-90: Avoid too much salt or sodium.
         1 - Not important at all
&        ,F4B       ! TYPE 50 VARIABLE  !    196      1   1989-90: Avoid too much saturated fat.
         1 - Not important at all
&        ,F4C1      ! TYPE 50 VARIABLE  !    197      1   1989-90: Eat at least five servings a day of
         1 - Not important at all
&        ,F4C2      ! TYPE 50 VARIABLE  !    198      1   1989-90: Blank, not asked.
         1 - Not important at all
&        ,F4C3      ! TYPE 50 VARIABLE  !    199      1   1989-90: Blank, not asked.
         1 - Not important at all
&        ,F4D       ! TYPE 50 VARIABLE  !    200      1   1989-90: Avoid too much sugar.
         1 - Not important at all
&        ,F4E1      ! TYPE 50 VARIABLE  !    201      1   1989:   Drink alcoholic beverages in
         1 - Not important at all
&        ,F4E2      ! TYPE 50 VARIABLE  !    202      1   1989:   Blank, not asked.
         1 - Not important at all
&        ,F4F       ! TYPE 50 VARIABLE  !    203      1   1989-90: Eat foods with adequate fiber.
         1 - Not important at all
&        ,F4G       ! TYPE 50 VARIABLE  !    204      1   1989-90: Eat foods with adequate starch.
         1 - Not important at all
&        ,F4H       ! TYPE 50 VARIABLE  !    205      1   1989-91: Eat a variety of foods.
         1 - Not important at all
&        ,F4I       ! TYPE 50 VARIABLE  !    206      1   1989-90: Maintain a desirable weight.
         1 - Not important at all
&        ,F4J       ! TYPE 50 VARIABLE  !    207      1   1989-90: Avoid too much fat.
         1 - Not important at all
&        ,F4K       ! TYPE 50 VARIABLE  !    208      1   1989-90: Avoid too much cholesterol.
         1 - Not important at all
&        ,F4L       ! TYPE 50 VARIABLE  !    209      1   1989-90: Eat at least six servings a day of
breads,  1 - Not important at all
&        ,F4M       ! TYPE 50 VARIABLE  !    210      1   1989-90: Blank, not asked.
         1 - Not important at all
&        ,F4N       ! TYPE 50 VARIABLE  !    211      1   1989:   Blank, not asked.
         1 - Not important at all
```

```
      &        ,F40         ! TYPE 50 VARIABLE  !   212   1  1989-90: Blank, not asked.
           1 - Not important at all
      IN_CNT = IN_CNT + 1
      TYPE_20_CNT = TYPE_20_CNT + 1


      IF(KFREQ .GT. 0) WRITE(8,1105) TYPE_20_CNT
1105  FORMAT(' TYPE_20_CNT =', I8)


      D1_FIRST = 0
      D1_LAST = 0
      D2_FIRST = 0
      D2_LAST = 0
      D3_FIRST = 0
      D3_LAST = 0

      NUM_30 = D1_NREC + D2_NREC + D3_NREC
      IF(NUM_30 .EQ. 0) THEN
         NUM_ZERO_30 = NUM_ZERO_30 + 1
         IF(NUM_ZERO_30 .LE. 5) THEN
            WRITE(PRFILE,*) '----- FOUND A ZERO FOOD REC PERSON ----'
            KFREQ = 1
         END IF
         GO TO 300
      END IF
      IF(KFREQ .GT. 0) WRITE(PRFILE,1130) NUM_30
1130  FORMAT(' NUM_30 =',I4)

      IF(D1_NREC .NE. 0) THEN
         D1_FIRST = 1
         D1_LAST  = D1_NREC
      END IF
      IF(D2_NREC .NE. 0) THEN
         D2_FIRST = D1_NREC + 1
         D2_LAST  = D1_NREC + D2_NREC
      END IF
      IF(D3_NREC .NE. 0) THEN
         D3_FIRST = D1_NREC + D2_NREC + 1
         D3_LAST  = D1_NREC + D2_NREC + D3_NREC
      END IF

C----------------READ TYPE 30 INFORMATION  -----
      DO J = 1,NUM_30
C         IF(KFREQ .GT. 0) WRITE(PRFILE,1140) J
C1140      FORMAT(' TYPE 30: J = ',I3)
         READ(9,END=900)
      &         RT_30(J)
      &        ,YEAR_30(J)
      &        ,SAMPLE_30(J)
      &        ,HHID_30(J)
      &        ,R_NUM_30(J)
      &        ,DAYCODE(J)
      &        ,SEQNUM(J)
      &        ,LINENUM(J)
      &        ,FOODCODE(J)
      &        ,FOODAMT(J)
      &        ,D_DAY(J)
      &        ,HOWEST(J)
      &        ,FAT_OCC(J)
      &        ,FAT_FOOD(J)
      &        ,FAT_TYPE(J)
      &        ,SALT_USE(J)
      &        ,CALEQ(J)
         IN_CNT = IN_CNT + 1
         TYPE_30_CNT = TYPE_30_CNT + 1
      END DO

         READ(9,END=900)
      &         RT_40
      &        ,YEAR_40
      &        ,SAMPLE_40
      &        ,HHID_40
      &        ,R_NUM_40
      &        ,RACE_40
      &        ,DAYCODE_40
```

31

```
      &           ,BMILK
      &           ,RDA1
      &           ,RDA2
      &           ,RDA3
      &           ,RDA4
      &           ,RDA5
      &           ,RDA6
      &           ,RDA7
      &           ,RDA8
      &           ,RDA9
      &           ,RDA10
      &           ,RDA11
      &           ,RDA12
      &           ,RDA13
      &           ,RDA14
      &           ,RDA15
      &           ,RDA16
      &           ,RDA17
      &           ,TOTNUT1
      &           ,TOTNUT2
      &           ,TOTNUT3
      &           ,TOTNUT4
      &           ,TOTNUT5
      &           ,TOTNUT6
      &           ,TOTNUT7
      &           ,TOTNUT8
      &           ,TOTNUT9
      &           ,TOTNUT10
      &           ,TOTNUT11
      &           ,TOTNUT29
            IN_CNT = IN_CNT + 1

300   CONTINUE
      RETURN


900   CONTINUE
C*************************************
C     EOF
C*************************************
      KEOF = 3

      WRITE(PRFILE,9010) IN_CNT
      &                 ,TYPE_20_CNT
      &                 ,TYPE_30_CNT
      &                 ,NUM_ZERO_30
9010  FORMAT(//
      &      ,/," IN_CNT  = ",I6
      &      ,/," TYPE_20_CNT   = ",I6
      &      ,/," TYPE_30_CNT   = ",I6
      &      ,/," NUM_ZERO_30   = ",I6
      &      )
      RETURN
      END



      SUBROUTINE PRINT_PERSON
C**************************************************
C--- READ IN A PERSON'S DATA
C**************************************************

      INTEGER*4 J
      CHARACTER*1 FF /Z'0C'/

      INCLUDE 'WORKCSF.INC'

C*****************************************
C------ PRINT A PERSON RECORD (TYPE 20)
C------ AND THEN ALL TYPE 30'S FOR
C------ THAT PERSON
C*****************************************


          WRITE(PRFILE,1120)
C--- TYPE 15
      &         RT_15     ! TYPE 15 VARIABLE
```

32

```
        &           ,YEAR_15   ! TYPE 15 VARIABLE
        &           ,SAMPLE_15 ! TYPE 15 VARIABLE
        &           ,HHID_15   ! TYPE 15 VARIABLE
        &           ,M_LGRADE  ! TYPE 15 VARIABLE
        &           ,F_LGRADE  ! TYPE 15 VARIABLE
        &           ,FOODSEC   ! TYPE 15 VARIABLE
        &           ,SURPLUS   ! TYPE 15 VARIABLE
C--- TYPE 20
        &           ,RT_20     ! TYPE 15-20 VARIABLE
        &           ,YEAR_20   ! TYPE 15-20 VARIABLE
        &           ,SAMPLE_20 ! TYPE 15-20 VARIABLE
        &           ,HHID_20   ! TYPE 15-20 VARIABLE
        &           ,R_NUM_20  ! TYPE 20 VARIABLE
        &           ,GEOG      ! TYPE 20 VARIABLE
        &           ,URB       ! TYPE 20 VARIABLE
        &           ,HHSZ      ! TYPE 20 VARIABLE
        &           ,INCOME    ! TYPE 20 VARIABLE
        &           ,PCTPOV    ! TYPE 20 VARIABLE
        &           ,H_RCVFS   ! TYPE 20 VARIABLE
        &           ,DHKHH     ! TYPE 20 VARIABLE
        &           ,CHILDREN  ! TYPE 20 VARIABLE
        &           ,HEADSTAT  ! TYPE 20 VARIABLE
        &           ,R_RCVFS   ! TYPE 20 VARIABLE
        &           ,AGE       ! TYPE 20 VARIABLE
        &           ,SEX       ! TYPE 20 VARIABLE
        &           ,PL_STAT   ! TYPE 20 VARIABLE
        &           ,RACE_20   ! TYPE 20 VARIABLE
        &           ,ORIGIN    ! TYPE 20 VARIABLE
        &           ,NUMDAYS   ! TYPE 20 VARIABLE
        &           ,R_WIC     ! TYPE 20 VARIABLE
        &           ,R_EMP     ! TYPE 20 VARIABLE
        &           ,R_LGRADE  ! TYPE 20 VARIABLE
        &           ,R_CNTL    ! TYPE 20 VARIABLE
        &           ,R_FREEL   ! TYPE 20 VARIABLE
        &           ,HLTHDIET  ! TYPE 20 VARIABLE
        &           ,SPECDIET  ! TYPE 20 VARIABLE
        &           ,DT_CALOR  ! TYPE 20 VARIABLE  !   136   1   Are you on a calorie/weight loss diet?
            1 - Yes
        &           ,DS_CALOR  ! TYPE 20 VARIABLE  !   137   1   Source of calorie/weight loss diet.
            1 - Doctor, dietician
        &           ,DT_FAT    ! TYPE 20 VARIABLE  !   138   1   Are you on a low fat/cholesterol diet?
            1 - Yes
        &           ,DS_FAT    ! TYPE 20 VARIABLE  !   139   1   Source of low fat/cholesterol diet.
            1 - Doctor, dietician
        &           ,DT_SALT   ! TYPE 20 VARIABLE  !   140   1   Are you on a low salt diet?
            1 - Yes
        &           ,DS_SALT   ! TYPE 20 VARIABLE  !   141   1   Source of low salt diet.
            1 - Doctor, dietician
        &           ,DT_SUGAR  ! TYPE 20 VARIABLE  !   142   1   Are you on a low sugar/sugar free diet?
            1 - Yes
        &           ,DS_SUGAR  ! TYPE 20 VARIABLE  !   143   1   Source of low sugar/sugar free diet.
            1 - Doctor, dietician
        &           ,DT_DIAB   ! TYPE 20 VARIABLE  !   144   1   Are you on a diabetic diet?
            1 - Yes
        &           ,DS_DIAB   ! TYPE 20 VARIABLE  !   145   1   Source of diabetic diet.
            1 - Doctor, dietician
        &           ,DT_LFIB   ! TYPE 20 VARIABLE  !   146   1   Are you on a low fiber diet?
            1 - Yes
        &           ,DS_LFIB   ! TYPE 20 VARIABLE  !   147   1   Source of low fiber diet.
            1 - Doctor, dietician
        &           ,DT_HFIB   ! TYPE 20 VARIABLE  !   148   1   Are you on a high fiber diet?
            1 - Yes
        &           ,DS_HFIB   ! TYPE 20 VARIABLE  !   149   1   Source of high fiber diet.
            1 - Doctor, dietician
        &           ,DT_OTHER  ! TYPE 20 VARIABLE  !   150   1   Are you on some other type of diet?
            1 - Yes
        &           ,DS_OTHER  ! TYPE 20 VARIABLE  !   151   1   Source of other diet.
            1 - Doctor, dietician
        &           ,VEGET     ! TYPE 20 VARIABLE  !   152   1   Do you consider yourself to be a vegetarian?
            1 - Yes
        &           ,R_WGT     ! TYPE 20 VARIABLE
        &           ,R_HGT     ! TYPE 20 VARIABLE
        &           ,BMI       ! TYPE 20 VARIABLE
        &           ,LEISURE   ! TYPE 20 VARIABLE  !   207   1   What would you say your usual level of
            1 - Heavy / rigorous
```

33

```
&        ,ACTIVE_L  ! TYPE 20 VARIABLE  !    208       1   Compared with most people your age and sex,
     1 - More active
&        ,D1_MNTH   ! TYPE 20 VARIABLE  ! 265-266      2   Date of intake: month.
     1 - January
&        ,D1_DATE   ! TYPE 20 VARIABLE  ! 267-268      2   Date of intake: day of the month.
     1 - 31
&        ,D1_YEAR   ! TYPE 20 VARIABLE  !    269       1   Date of intake: year.
     9 - 1989
&        ,D1_DAY    ! TYPE 20 VARIABLE  !    270       1   Date of intake: day of the week.
     1 - Sunday
&        ,D1_NREC   ! TYPE 20 VARIABLE
&        ,D1_FEW    ! TYPE 20 VARIABLE  !    273       1   Explanation for few or no food records.
     3 - Ill, no food or beverage
&        ,D1_AMTUS  ! TYPE 20 VARIABLE  !    281       1   Would you say that the amount of food and drink
     1 - Less than usual
&        ,D1_RESUS  ! TYPE 20 VARIABLE  !    282       1   What best describes the reason for the
     1 - Sick or ill
&        ,D2_MNTH   ! TYPE 20 VARIABLE  ! 285-286      2   Date of intake: month.
     1 - January
&        ,D2_DATE   ! TYPE 20 VARIABLE  ! 287-288      2   Date of intake: day of the month.
     1 - 31
&        ,D2_YEAR   ! TYPE 20 VARIABLE  !    289       1   Date of intake: year.
     9 - 1989
&        ,D2_DAY    ! TYPE 20 VARIABLE  !    290       1   Date of intake: day of the week.
     1 - Sunday
&        ,D2_NREC   ! TYPE 20 VARIABLE
&        ,D2_FEW    ! TYPE 20 VARIABLE  !    293       1   Explanation for few or no food records.
     3 - Ill, no food or beverage
&        ,D2_AMTUS  ! TYPE 20 VARIABLE  !    301       1   Would you say that the amount of food and drink
     1 - Less than usual
&        ,D2_RESUS  ! TYPE 20 VARIABLE  !    302       1   What best describes the reason for the
     1 - Sick or ill
&        ,D3_MNTH   ! TYPE 20 VARIABLE  ! 306-307      2   Date of intake: month.
     1 - January
&        ,D3_DATE   ! TYPE 20 VARIABLE  ! 308-309      2   Date of intake: day of the month.
     1 - 31
&        ,D3_YEAR   ! TYPE 20 VARIABLE  !    310       1   Date of intake: year.
     9 - 1989
&        ,D3_DAY    ! TYPE 20 VARIABLE  !    311       1   Date of intake: day of the week.
     1 - Sunday
&        ,D3_NREC   ! TYPE 20 VARIABLE
&        ,D3_FEW    ! TYPE 20 VARIABLE  !    314       1   Explanation for few or no food records.
     3 - Ill, no food or beverage
&        ,D3_AMTUS  ! TYPE 20 VARIABLE  !    322       1   Would you say that the amount of food and drink
     1 - Less than usual
&        ,D3_RESUS  ! TYPE 20 VARIABLE  !    323       1   What best describes the reason for the
     1 - Sick or ill
&        ,WGT_3DAY  ! TYPE 20 VARIABLE  ! 578-580      3
C--- TYPE 50
&        ,RT_50     ! TYPE 50 VARIABLE  !    1-2       2   Record type.
     50 - All records
&        ,YEAR_50   ! TYPE 50 VARIABLE  !     3        1   Year of survey.
     1 - 1989
&        ,SAMPLE_50 ! TYPE 50 VARIABLE  !     4        1   Sample.
     1 - Basic
&        ,HHID_50   ! TYPE 50 VARIABLE  !    5-9       5   Household identification number.
&        ,R_NUM_50  ! TYPE 50 VARIABLE  !   10-11      2   Line number of DHKS respondent.
&        ,RACE_50   ! TYPE 50 VARIABLE  !    66        1   Race.
     1 - White
&        ,F1A       ! TYPE 50 VARIABLE  ! 153-154      2   1989:   Blank, not asked.
     0 - 97
&        ,F1B       ! TYPE 50 VARIABLE  ! 155-156      2   1989:   Blank, not asked.
     0 - 97
&        ,F1C       ! TYPE 50 VARIABLE  ! 157-158      2   1989:   Blank, not asked.
     0 - 97
&        ,F1D       ! TYPE 50 VARIABLE  ! 159-160      2   1989:   Blank, not asked.
     0 - 97
&        ,F1E       ! TYPE 50 VARIABLE  ! 161-162      2   1989:   Blank, not asked.
     0 - 97
&        ,F4A       ! TYPE 50 VARIABLE  !    195       1   1989-90: Avoid too much salt or sodium.
     1 - Not important at all
&        ,F4B       ! TYPE 50 VARIABLE  !    196       1   1989-90: Avoid too much saturated fat.
     1 - Not important at all
&        ,F4C1      ! TYPE 50 VARIABLE  !    197       1   1989-90: Eat at least five servings a day of
     1 - Not important at all
```

```
     &       ,F4C2      ! TYPE 50 VARIABLE  !   198    1   1989-90: Blank, not asked.
             1 - Not important at all
     &       ,F4C3      ! TYPE 50 VARIABLE  !   199    1   1989-90: Blank, not asked.
             1 - Not important at all
     &       ,F4D       ! TYPE 50 VARIABLE  !   200    1   1989-90: Avoid too much sugar.
             1 - Not important at all
     &       ,F4E1      ! TYPE 50 VARIABLE  !   201    1   1989:   Drink alcoholic beverages in
             1 - Not important at all
     &       ,F4E2      ! TYPE 50 VARIABLE  !   202    1   1989:   Blank, not asked.
             1 - Not important at all
     &       ,F4F       ! TYPE 50 VARIABLE  !   203    1   1989-90: Eat foods with adequate fiber.
             1 - Not important at all
     &       ,F4G       ! TYPE 50 VARIABLE  !   204    1   1989-90: Eat foods with adequate starch.
             1 - Not important at all
     &       ,F4H       ! TYPE 50 VARIABLE  !   205    1   1989-91: Eat a variety of foods.
             1 - Not important at all
     &       ,F4I       ! TYPE 50 VARIABLE  !   206    1   1989-90: Maintain a desirable weight.
             1 - Not important at all
     &       ,F4J       ! TYPE 50 VARIABLE  !   207    1   1989-90: Avoid too much fat.
             1 - Not important at all
     &       ,F4K       ! TYPE 50 VARIABLE  !   208    1   1989-90: Avoid too much cholesterol.
             1 - Not important at all
     &       ,F4L       ! TYPE 50 VARIABLE  !   209    1   1989-90: Eat at least six servings a day of
breads,      1 - Not important at all
     &       ,F4M       ! TYPE 50 VARIABLE  !   210    1   1989-90: Blank, not asked.
             1 - Not important at all
     &       ,F4N       ! TYPE 50 VARIABLE  !   211    1   1989:   Blank, not asked.
             1 - Not important at all
     &       ,F4O       ! TYPE 50 VARIABLE  !   212    1   1989-90: Blank, not asked.
             1 - Not important at all
1120     FORMAT(
     &           //'---TYPE 15 INFOMATION---'
     &           ,/,' RT_15   =',A8
     &           ,' YEAR_15  =',I8
     &           ,' SAMPLE_15=',I8
     &           ,' HHID_15  =',I8
     &           /,' M_LGRADE =',I8
     &           ,' F_LGRADE =',I8
     &           ,' FOODSEC  =',I8
     &           ,' SURPLUS  =',I8
     &           ,//,'---TYPE 20 INFOMATION---'
     &           ,/,' RT_20   =',A8
     &           ,' YEAR_20  =',I8
     &           ,' SAMPLE_20=',I8
     &           ,' HHID_20  =',I8
     &           ,' R_NUM_20 =',I8
     &           ,/,' GEOG    =',I8
     &           ,' URB     =',I8
     &           ,' HHSZ    =',I8
     &           ,' INCOME  =',I8
     &           ,' PCTPOV  =',I8
     &           ,/,' H_RCVFS =',I8
     &           ,' DHKHH   =',I8
     &           ,' CHILDREN =',I8
     &           ,' HEADSTAT =',I8
     &           ,' R_RCVFS =',I8
     &           ,/,' AGE     =',I8
     &           ,' SEX     =',I8
     &           ,' PL_STAT =',I8
     &           ,' RACE_20 =',I8
     &           ,' ORIGIN  =',I8
     &           ,/,' NUMDAYS =',I8
     &           ,' R_WIC   =',I8
     &           ,' R_EMP   =',I8
     &           ,' R_LGRADE =',I8
     &           ,' R_CNTL  =',I8
     &           ,' R_FREEL =',I8
     &           ,/,' HLTHDIET =',I8
     &           ,' SPECDIET =',I8
     &           ,' DT_CALOR =',I8
     &           ,' DS_CALOR =',I8
     &           ,' DT_FAT  =',I8
     &           ,' DS_FAT  =',I8
     &           ,/,' DT_SALT =',I8
     &           ,' DS_SALT =',I8
     &           ,' DT_SUGAR =',I8
```

35

```
&             ,' DS_SUGAR =',I8
&             ,' DT_DIAB  =',I8
&             ,' DS_DIAB  =',I8
&          ,/,' DT_LFIB  =',I8
&             ,' DS_LFIB  =',I8
&             ,' DT_HFIB  =',I8
&             ,' DS_HFIB  =',I8
&             ,' DT_OTHER =',I8
&             ,' DS_OTHER =',I8
&             ,' VEGET    =',I8
&          ,/,' R_WGT    =',I8
&             ,' R_HGT    =',I8
&             ,' BMI      =',F8.2
&             ,' LEISURE  =',I8
&             ,' ACTIVE_L =',I8
&          ,/,' D1_MNTH  =',I8
&             ,' D1_DATE  =',I8
&             ,' D1_YEAR  =',I8
&             ,' D1_DAY   =',I8
&          ,/,' D1_NREC  =',I8
&             ,' D1_FEW   =',I8
&             ,' D1_AMTUS =',I8
&             ,' D1_RESUS =',I8
&          ,/,' D2_MNTH  =',I8
&             ,' D2_DATE  =',I8
&             ,' D2_YEAR  =',I8
&             ,' D2_DAY   =',I8
&          ,/,' D2_NREC  =',I8
&             ,' D2_FEW   =',I8
&             ,' D2_AMTUS =',I8
&             ,' D2_RESUS =',I8
&          ,/,' D3_MNTH  =',I8
&             ,' D3_DATE  =',I8
&             ,' D3_YEAR  =',I8
&             ,' D3_DAY   =',I8
&          ,/,' D3_NREC  =',I8
&             ,' D3_FEW   =',I8
&             ,' D3_AMTUS =',I8
&             ,' D3_RESUS =',I8
&          ,/,' WGT_3DAY =',I8
&          ,//,'---TYPE 50 INFOMATION---'
&          ,/,' RT_50    =',A8
&             ,' YEAR_50  =',I8
&             ,' SAMPLE_50=',I8
&             ,' HHID_50  =',I8
&             ,' R_NUM_50 =',I8
&             ,' RACE_50  =',I8
&          ,/,' F1A      =',I8
&             ,' F1B      =',I8
&             ,' F1C      =',I8
&             ,' F1D      =',I8
&             ,' F1E      =',I8
&          ,/,' F4A      =',I8
&             ,' F4B      =',I8
&             ,' F4C1     =',I8
&             ,' F4C2     =',I8
&             ,' F4C3     =',I8
&          ,/,' F4D      =',I8
&             ,' F4E1     =',I8
&             ,' F4E2     =',I8
&             ,' F4F      =',I8
&             ,' F4G      =',I8
&          ,/,' F4H      =',I8
&             ,' F4I      =',I8
&             ,' F4J      =',I8
&             ,' F4K      =',I8
&             ,' F4L      =',I8
&          ,/,' F4M      =',I8
&             ,' F4N      =',I8
&             ,' F4O      =',I8
&             )

      WRITE(PRFILE,2220)
&          RT_40
&          ,YEAR_40
&          ,SAMPLE_40
```

36

```fortran
     &          ,HHID_40
     &          ,R_NUM_40
     &          ,RACE_40
     &          ,DAYCODE_40
     &          ,BMILK
     &          ,RDA1
     &          ,RDA2
     &          ,RDA3
     &          ,RDA4
     &          ,RDA5
     &          ,RDA6
     &          ,RDA7
     &          ,RDA8
     &          ,RDA9
     &          ,RDA10
     &          ,RDA11
     &          ,RDA12
     &          ,RDA13
     &          ,RDA14
     &          ,RDA15
     &          ,RDA16
     &          ,RDA17
     &          ,TOTNUT1
     &          ,TOTNUT2
     &          ,TOTNUT3
     &          ,TOTNUT4
     &          ,TOTNUT5
     &          ,TOTNUT6
     &          ,TOTNUT7
     &          ,TOTNUT8
     &          ,TOTNUT9
     &          ,TOTNUT10
     &          ,TOTNUT11
     &          ,TOTNUT29
2220     FORMAT(
     &      /,'---TYPE 40 INFOMATION---'
     &    ,/,' RT_40     =',A8
     &      ,' YEAR_40   =',I8
     &      ,' SAMPLE_40 =',I8
     &    ,/,' HHID_40   =',I8
     &      ,' R_NUM_40  =',I8
     &      ,' RACE_40   =',I8
     &      ,' DAYCODE_40=',I8
     &      ,' BMILK     =',I8
     &    ,/,' RDA1      =',F8.2
     &      ,' RDA2      =',F8.2
     &      ,' RDA3      =',F8.2
     &      ,' RDA4      =',F8.2
     &      ,' RDA5      =',F8.2
     &    ,/,' RDA6      =',F8.2
     &      ,' RDA7      =',F8.2
     &      ,' RDA8      =',F8.2
     &      ,' RDA9      =',F8.2
     &      ,' RDA10     =',F8.2
     &    ,/,' RDA11     =',F8.2
     &      ,' RDA12     =',F8.2
     &      ,' RDA13     =',F8.2
     &      ,' RDA14     =',F8.2
     &      ,' RDA15     =',F8.2
     &      ,' RDA16     =',F8.2
     &      ,' RDA17     =',F8.2
     &    ,/,' TOTNUT1   =',F8.2
     &      ,' TOTNUT2   =',F8.2
     &      ,' TOTNUT3   =',F8.2
     &      ,' TOTNUT4   =',F8.2
     &      ,' TOTNUT5   =',F8.2
     &    ,/,' TOTNUT6   =',F8.2
     &      ,' TOTNUT7   =',F8.2
     &      ,' TOTNUT8   =',F8.2
     &      ,' TOTNUT9   =',F8.2
     &      ,' TOTNUT10  =',F8.2
     &    ,/,' TOTNUT11  =',F8.2
     &      ,' TOTNUT29  =',F8.2
     &      )
C----------------------------------------
C  DAY 1 FOOD ITEMS
```

37

```
C--------------------------------------------
       IF(D1_NREC .EQ. 0) THEN
          GO TO 200
       END IF
       IF(KFREQ .GT. 5) WRITE(PRFILE,1130) D1_FIRST,D1_LAST
1130   FORMAT(/,' DAY 1: D1_FIRST = ',I3,'  D1_LAST = ',I3)
C----------PRINT TYPE 30 INFORMATION  -----
       WRITE(PRFILE,1210)
1210   FORMAT(
    &              /,'   REC_NUM'
    &              ,'   DAYCODE'
    &              ,'    SEQNUM'
    &              ,'   LINENUM'
    &              ,'  FOODCODE'
    &              ,'   FOODAMT'
    &              ,'     D_DAY'
    &              ,'    HOWEST'
    &              ,'   FAT_OCC'
    &              ,'  FAT_FOOD'
    &              ,'  FAT_TYPE'
    &              ,'  SALT_USE'
    &              ,'     CALEQ'
    &              )
       DO J = D1_FIRST,D1_LAST
          WRITE(PRFILE,1220)
    &              J
    &              ,DAYCODE(J)
    &              ,SEQNUM(J)
    &              ,LINENUM(J)
    &              ,FOODCODE(J)
    &              ,FOODAMT(J)
    &              ,D_DAY(J)
    &              ,HOWEST(J)
    &              ,FAT_OCC(J)
    &              ,FAT_FOOD(J)
    &              ,FAT_TYPE(J)
    &              ,SALT_USE(J)
    &              ,CALEQ(J)
1220   FORMAT(
    &              I10
    &              ,I10
    &              ,I10
    &              ,I10
    &              ,I10
    &              ,F10.2
    &              ,I10
    &              ,I10
    &              ,I10
    &              ,I10
    &              ,I10
    &              ,I10
    &              ,F10.2
    &              )
       END DO

200   CONTINUE
C----------------------------------------
C  DAY 2 FOOD ITEMS
C----------------------------------------
       IF(D2_NREC .EQ. 0) THEN
          GO TO 300
       END IF
       IF(KFREQ .GT. 5) WRITE(PRFILE,2110) D2_FIRST,D2_LAST
2110   FORMAT(/,' DAY 2: D2_FIRST = ',I3,'  D2_LAST = ',I3)
       WRITE(PRFILE,1210)
       DO J = D2_FIRST,D2_LAST
          WRITE(PRFILE,1220)
    &              J
    &              ,DAYCODE(J)
    &              ,SEQNUM(J)
    &              ,LINENUM(J)
    &              ,FOODCODE(J)
    &              ,FOODAMT(J)
    &              ,D_DAY(J)
    &              ,HOWEST(J)
    &              ,FAT_OCC(J)
```

38

```
      &               ,FAT_FOOD(J)
      &               ,FAT_TYPE(J)
      &               ,SALT_USE(J)
      &               ,CALEQ(J)
          END DO

300   CONTINUE
C----------------------------------------
C  DAY 3 FOOD ITEMS
C----------------------------------------
      IF(D3_NREC .EQ. 0) THEN
          GO TO 400
      END IF
      IF(KFREQ .GT. 5) WRITE(PRFILE,3110) D3_FIRST,D3_LAST
3110  FORMAT(/,' DAY 3: D3_FIRST = ',I3,'  D3_LAST = ',I3)
      WRITE(PRFILE,1210)  ! the header line
      DO J = D3_FIRST,D3_LAST
          WRITE(PRFILE,1220)
      &              J
      &              ,DAYCODE(J)
      &              ,SEQNUM(J)
      &              ,LINENUM(J)
      &              ,FOODCODE(J)
      &              ,FOODAMT(J)
      &              ,D_DAY(J)
      &              ,HOWEST(J)
      &              ,FAT_OCC(J)
      &              ,FAT_FOOD(J)
      &              ,FAT_TYPE(J)
      &              ,SALT_USE(J)
      &              ,CALEQ(J)
          END DO
      WRITE(PRFILE,*) FF   ! PERFORM A PAGE EJECT

400   CONTINUE
      RETURN
      END


      SUBROUTINE CREATE_INDEX
C***************************************************
C--- EXTRACT A PERSON'S DATA
C***************************************************
      IMPLICIT NONE
      REAL*4  PS1
      REAL*4  PS2
      REAL*4  PS3
      REAL*4  PS4
      REAL*4  PS5
      REAL*4  LEGUME
      REAL*4  SCALE1
      REAL*4  SCALE2
      REAL*4  SCALE3
      REAL*4  SCALE4
      REAL*4  SCALE5
      REAL*4  SCALE6
      REAL*4  MAXPOINT
      REAL*4  MINPOINT
      REAL*4  FOOD_PORTION

      REAL*4 CC1A
      REAL*4 CC1B
      REAL*4 CC2A
      REAL*4 CC2B
      REAL*4 CC2C
      REAL*4 CC3A
      REAL*4 CC3B
      REAL*4 CC4A
      REAL*4 CC5A
      REAL*4 CC5B
      REAL*4 CC5C
      REAL*4 CC6A
      REAL*4 CC6B

      INTEGER*4 SPLIT_FOODCODE(500)
      REAL*4    SPLIT_PORTION(500)
```

```
      INTEGER*4 AGG_FOODCODE(500)
      REAL*4    AGG_PORTION(500)
      INTEGER*4 VARIETY_CNT

      REAL*4 NUM_DAYS
      LOGICAL*4 PROCESS_FOODCODE(500)

      INTEGER*4 I
      INTEGER*4 J
      INTEGER*4 PICK
      INTEGER*4 IAGE
      INTEGER*4 ISEX
      INTEGER*4 INVALID_SEX_CNT/0/
      INTEGER*4 VERY_OLD_AGE_CNT/0/
      INTEGER*4 LEGUME_CNT/0/

      INCLUDE 'WORKCSF.INC'
      INCLUDE 'INDEX.INC'
      INCLUDE 'F7_CODE.INC'

      IF(KFREQ .GT. 0) WRITE(PRFILE,*) ' ---- IN CREATE INDEX ----'

C---- CREATE NEW HEAD'S GRADE COMPLETED VARIABLE
      IF(M_LGRADE .NE. 99 .AND. M_LGRADE .GE. F_LGRADE) THEN
          LGRADE_15 = M_LGRADE
      ELSE IF(F_LGRADE .NE. 99 .AND. F_LGRADE .GT. M_LGRADE) THEN
          LGRADE_15 = F_LGRADE
      ELSE
          LGRADE_15 = 99
      END IF

      IF(KFREQ .GT. 0) WRITE(PRFILE,1010) LGRADE_15,M_LGRADE,F_LGRADE
1010  FORMAT(' LGRADE_15 =',I4,'  M_LGRADE =',I4,'  F_LGRADE =',I4)

      FG1 = 0.0
      FG2 = 0.0
      FG3 = 0.0
      FG4 = 0.0
      FG5 = 0.0
      FG6 = 0.0

      AVG_FG1 = 0.0
      AVG_FG2 = 0.0
      AVG_FG3 = 0.0
      AVG_FG4 = 0.0
      AVG_FG5 = 0.0
      AVG_LEGUME = 0.0

      NUM_DAYS = 0.0

      COMP01 = 0.0
      COMP02 = 0.0
      COMP03 = 0.0
      COMP04 = 0.0
      COMP05 = 0.0
      COMP06 = 0.0
      COMP07 = 0.0
      COMP08 = 0.0
      COMP09 = 0.0
      COMP10 = 0.0

      SPLIT_CNT = 0
      AGG_CNT = 0
      HUMAN_MILK = .FALSE.

      IF(AGE .GT. 3) THEN ! SCALE YOUNG KIDS PORTIONS UP
          SCALE1 = 1.0
          SCALE2 = 1.0
          SCALE3 = 1.0
          SCALE4 = 1.0
          SCALE5 = 1.0
          SCALE6 = 1.0
      ELSE
          SCALE1 = 6.0 / 4.9
          SCALE2 = 3.0 / 2.5
          SCALE3 = 2.0 / 1.5
```

```
              SCALE4 = 1.0
              SCALE5 = 3.0 / 2.5 ! NB meats need to be converted from oz to servings
              SCALE6 = 3.0 / 2.5
          END IF
          IF(KFREQ .GT. 0) THEN
            WRITE(PRFILE,1020)
     &               SCALE1
     &              ,SCALE2
     &              ,SCALE3
     &              ,SCALE4
     &              ,SCALE5
     &              ,SCALE6
1020      FORMAT(
     &               ' SCALE1 =',F6.3
     &              ,' SCALE2 =',F6.3
     &              ,' SCALE3 =',F6.3
     &              ,' SCALE4 =',F6.3
     &              ,' SCALE5 =',F6.3
     &              ,' SCALE6 =',F6.3
     &           )
          END IF

          IF(NUM_30 .EQ. 0) GO TO 900   ! no food records

C----- initialize the split-foodcode arrays
          DO I = 1,500
             SPLIT_FOODCODE(I) = 0
             SPLIT_PORTION(I) = 0.0
             AGG_FOODCODE(I) = 0
             AGG_PORTION(I) = 0.0
          END DO

C----- set up a logical to tell us whether or not to process the record
          DO I = 1,NUM_30
             PROCESS_FOODCODE(I) = .FALSE.
             IF(DAY1 .AND. DAYCODE(I) .EQ. 1) THEN
                PROCESS_FOODCODE(I) = .TRUE.
             END IF
             IF(DAY2 .AND. DAYCODE(I) .EQ. 2) THEN
                PROCESS_FOODCODE(I) = .TRUE.
             END IF
             IF(DAY3 .AND. DAYCODE(I) .EQ. 3) THEN
                PROCESS_FOODCODE(I) = .TRUE.
             END IF
          END DO


C---- count days that the person has data for the
C---- the days we want to process
          IF(D1_NREC .GT. 0 .AND. DAY1) THEN
             NUM_DAYS = NUM_DAYS + 1
          END IF
          IF(D2_NREC .GT. 0 .AND. DAY2) THEN
             NUM_DAYS = NUM_DAYS + 1
          END IF
          IF(D3_NREC .GT. 0 .AND. DAY3) THEN
             NUM_DAYS = NUM_DAYS + 1
          END IF
          IF(KFREQ .GT. 0) THEN
            WRITE(PRFILE,1030) NUMDAYS
     &                         ,D1_NREC
     &                         ,D2_NREC
     &                         ,D3_NREC
1030      FORMAT(' -- starting loop over foodcodes --'
     &          ,' NUMDAYS =', I2
     &          ,' D1_NREC =', I2
     &          ,' D2_NREC =', I2
     &          ,' D3_NREC =', I2
     &          )
          END IF

          DO I = 1,NUM_30
             IF( .NOT. PROCESS_FOODCODE(I)) CYCLE
             IF(FOODCODE(I) .EQ. 1100000) THEN
                HUMAN_MILK = .TRUE.
                HUMAN_MILK_CNT = HUMAN_MILK_CNT + 1        41
```

```fortran
          GO TO 900
      END IF

      CALL BINARY_SEARCH_INT4(FOODCODE(I),PICK,1
     &                        ,MAX_CODES,KFREQ)
      FOODLABEL(I) = LABEL7(PICK)
      CC1A = VFC(PICK,01)
      CC1B = VFC(PICK,02)
      CC2A = VFC(PICK,03)
      CC2B = VFC(PICK,04)
      CC2C = VFC(PICK,05)
      CC3A = VFC(PICK,06)
      CC3B = VFC(PICK,07)
      CC4A = VFC(PICK,08)
      CC5A = VFC(PICK,09)
      CC5B = VFC(PICK,10)
      CC5C = VFC(PICK,11)
      CC6A = VFC(PICK,12)
      CC6B = VFC(PICK,13)
      PS1 =  SERVINGS(PICK,1) * (FOODAMT(I) / 100.0) * SCALE1      ! GRAINS
      PS2 =  SERVINGS(PICK,2) * (FOODAMT(I) / 100.0) * SCALE2      ! VEGGIES
      PS3 =  SERVINGS(PICK,3) * (FOODAMT(I) / 100.0) * SCALE3      ! FRUIT
      PS4 =  SERVINGS(PICK,4) * (FOODAMT(I) / 100.0) * SCALE4      ! DAIRY
      PS5 =  SERVINGS(PICK,5)/2.5 * (FOODAMT(I)/100.0) * SCALE5     ! MEATS
      LEGUME =  SERVINGS(PICK,6) * (FOODAMT(I) / 100.0) * SCALE6  ! legumes
      FOOD_PORTION = PS1 + PS2 + PS3 + PS4 + PS5 + LEGUME

      IF(KFREQ .GT. 0) THEN
         WRITE(PRFILE,1110) PICK,FOODAMT(I)
     &                      ,FOODCODE(I),FOODLABEL(I)
     &                      ,PS1,SERVINGS(PICK,1)
     &                      ,PS2,SERVINGS(PICK,2)
     &                      ,PS3,SERVINGS(PICK,3)
     &                      ,PS4,SERVINGS(PICK,4)
     &                      ,PS5,SERVINGS(PICK,5)
     &                      ,LEGUME,SERVINGS(PICK,6)
     &                      ,FOOD_PORTION
     &                      ,CC1A,CC1B
     &                      ,CC2A,CC2B,CC2C
     &                      ,CC3A,CC3B
     &                      ,CC4A
     &                      ,CC5A,CC5B,CC5C
     &                      ,CC6A,CC6B
1110     FORMAT(
     &        / '  PICK      = ',I6
     &         ,'  FOODAMT = ',F7.2
     &         ,'  FOODCODE = ',I7
     &         ,': ',A40
     &        /,'  PS1       = ',F6.2
     &         ,'  SERVINGS = ',F6.2
     &        /,'  PS2       = ',F6.2
     &         ,'  SERVINGS = ',F6.2
     &        /,'  PS3       = ',F6.2
     &         ,'  SERVINGS = ',F6.2
     &        /,'  PS4       = ',F6.2
     &         ,'  SERVINGS = ',F6.2
     &        /,'  PS5       = ',F6.2
     &         ,'  SERVINGS = ',F6.2
     &        /,'  LEGUME    = ',F6.2
     &         ,'  SERVINGS = ',F6.2
     &        /,'  FOOD PORTION = ',F6.2
     &        /,'  CC1A = ',I8,'  CC1B = ',I8
     &        /,'  CC2A = ',I8,'  CC2B = ',I8,'  CC2C = ',I8
     &        /,'  CC3A = ',I8,'  CC3B = ',I8
     &        /,'  CC4A = ',I8
     &        /,'  CC5A = ',I8,'  CC5B = ',I8,'  CC5C = ',I8
     &        /,'  CC6A = ',I8,'  CC6B = ',I8
     &            )
      END IF

      FG1 = FG1 + PS1
      FG2 = FG2 + PS2
      FG3 = FG3 + PS3
      FG4 = FG4 + PS4
      FG5 = FG5 + PS5
      FG6 = FG6 + LEGUME
```

42

```fortran
C--- create an array of "split out" food types
          IF(FOOD_PORTION .GT. 0.0) THEN
              IF(CC1A .GT. 0 .AND. CC1B .GT. 0) THEN
                  SPLIT_CNT = SPLIT_CNT + 1
                  SPLIT_FOODCODE(SPLIT_CNT) = CC1A
                  SPLIT_PORTION(SPLIT_CNT) = PS1/2.0
                  SPLIT_CNT = SPLIT_CNT + 1
                  SPLIT_FOODCODE(SPLIT_CNT) = CC1B
                  SPLIT_PORTION(SPLIT_CNT) = PS1/2.0
              ELSE IF(CC1A .GT. 0) THEN
              ! I assume b can't be > 0 if a is not > 0
                  SPLIT_CNT = SPLIT_CNT + 1
                  SPLIT_FOODCODE(SPLIT_CNT) = CC1A
                  SPLIT_PORTION(SPLIT_CNT) = PS1
              END IF
              IF(CC2A .GT. 0 .AND. CC2B .GT. 0) THEN
                  SPLIT_CNT = SPLIT_CNT + 1
                  SPLIT_FOODCODE(SPLIT_CNT) = CC2A
                  SPLIT_PORTION(SPLIT_CNT) = PS2/2.0
                  SPLIT_CNT = SPLIT_CNT + 1
                  SPLIT_FOODCODE(SPLIT_CNT) = CC2B
                  SPLIT_PORTION(SPLIT_CNT) = PS2/2.0
              ELSE IF(CC2A .GT. 0) THEN
              ! I assume b can't be > 0 if a is not > 0
                  SPLIT_CNT = SPLIT_CNT + 1
                  SPLIT_FOODCODE(SPLIT_CNT) = CC2A
                  SPLIT_PORTION(SPLIT_CNT) = PS2
              END IF
              IF(CC3A .GT. 0 .AND. CC3B .GT. 0) THEN
                  SPLIT_CNT = SPLIT_CNT + 1
                  SPLIT_FOODCODE(SPLIT_CNT) = CC3A
                  SPLIT_PORTION(SPLIT_CNT) = PS3/2.0
                  SPLIT_CNT = SPLIT_CNT + 1
                  SPLIT_FOODCODE(SPLIT_CNT) = CC3B
                  SPLIT_PORTION(SPLIT_CNT) = PS3/2.0
              ELSE IF(CC3A .GT. 0) THEN
              ! I assume b can't be > 0 if a is not > 0
                  SPLIT_CNT = SPLIT_CNT + 1
                  SPLIT_FOODCODE(SPLIT_CNT) = CC3A
                  SPLIT_PORTION(SPLIT_CNT) = PS3
              END IF
              IF(CC4A .GT. 0) THEN ! milk has only one category
                  SPLIT_CNT = SPLIT_CNT + 1
                  SPLIT_FOODCODE(SPLIT_CNT) = CC4A
                  SPLIT_PORTION(SPLIT_CNT) = PS4
              END IF
              IF(CC5A .GT. 0 .AND. CC5B .GT. 0) THEN
                  SPLIT_CNT = SPLIT_CNT + 1
                  SPLIT_FOODCODE(SPLIT_CNT) = CC5A
                  SPLIT_PORTION(SPLIT_CNT) = PS5/2.0
                  SPLIT_CNT = SPLIT_CNT + 1
                  SPLIT_FOODCODE(SPLIT_CNT) = CC5B
                  SPLIT_PORTION(SPLIT_CNT) = PS5/2.0
              ELSE IF(CC5A .GT. 0) THEN
              ! I assume b can't be > 0 if a is not > 0
                  SPLIT_CNT = SPLIT_CNT + 1
                  SPLIT_FOODCODE(SPLIT_CNT) = CC5A
                  SPLIT_PORTION(SPLIT_CNT) = PS5
              END IF
              IF(CC6A .GT. 0 .AND. CC6B .GT. 0) THEN
                  SPLIT_CNT = SPLIT_CNT + 1
                  SPLIT_FOODCODE(SPLIT_CNT) = CC6A
                  SPLIT_PORTION(SPLIT_CNT) = LEGUME/2.0
                  SPLIT_CNT = SPLIT_CNT + 1
                  SPLIT_FOODCODE(SPLIT_CNT) = CC6B
                  SPLIT_PORTION(SPLIT_CNT) = LEGUME/2.0
              ELSE IF(CC6A .GT. 0) THEN
              ! I assume b can't be > 0 if a is not > 0
                  SPLIT_CNT = SPLIT_CNT + 1
                  SPLIT_FOODCODE(SPLIT_CNT) = CC6A
                  SPLIT_PORTION(SPLIT_CNT) = LEGUME
              END IF

          END IF  ! food portion gt 0.0
       END DO  ! number of food items
```

43

```
C----   aggregate the split foodcodes
        IF(SPLIT_CNT .EQ. 0) THEN
           WRITE(6,*) ' ** ERROR: FOUND SPLIT_CNT = 0 '
           WRITE(PRFILE,*) ' ** ERROR FOUND SPLIT_CNT = 0 **'
           WRITE(PRFILE,*) ' HHID = ', HHID_20
           KFREQ = 1
           GO TO 900
        END IF

        AGG_FOODCODE(1) = SPLIT_FOODCODE(1)
        AGG_PORTION(1) = SPLIT_PORTION(1)
        AGG_CNT = 1
        DO I = 2,SPLIT_CNT
           DO J = 1,AGG_CNT
             IF(AGG_FOODCODE(J) .EQ. SPLIT_FOODCODE(I)) THEN
                AGG_PORTION(J) = AGG_PORTION(J) + SPLIT_PORTION(I)
                GO TO 210
             END IF
           END DO ! over j
           ! get to here if no match found; so add to agg list
           AGG_CNT = AGG_CNT + 1
           AGG_FOODCODE(AGG_CNT) = SPLIT_FOODCODE(I)
           AGG_PORTION(AGG_CNT)  = SPLIT_PORTION(I)
210        CONTINUE ! come to here if inner loop found a match
        END DO ! over split_cnt

C-----  print the unaggregated split foodcodes
        IF(KFREQ .GT. 0) THEN
           WRITE(PRFILE,*) ' --- PRINT OF UNAGREGATED SPLIT FOODCODES'
           WRITE(PRFILE,*) ' SPLIT_CNT = ', SPLIT_CNT
           DO I = 1,SPLIT_CNT
              WRITE(PRFILE,1210) SPLIT_FOODCODE(I),SPLIT_PORTION(I)
1210          FORMAT(' foodcode = ',I7,'   PORTION = ',F6.3)
           END DO

C-----  print the aggregated split foodcodes
           WRITE(PRFILE,*) ' '
           WRITE(PRFILE,*) ' --- PRINT OF AGREGATED SPLIT FOODCODES'
           WRITE(PRFILE,*) ' AGG_CNT = ', AGG_CNT
           DO I = 1,AGG_CNT
              WRITE(PRFILE,1220) AGG_FOODCODE(I),AGG_PORTION(I)
1220          FORMAT(' foodcode = ',I7,'   PORTION = ',F6.3)
           END DO
        END IF ! kfreq



        IF(NUM_DAYS .GT. 0) THEN
           AVG_FG1 = FG1 / NUM_DAYS
           AVG_FG2 = FG2 / NUM_DAYS
           AVG_FG3 = FG3 / NUM_DAYS
           AVG_FG4 = FG4 / NUM_DAYS
           AVG_FG5 = FG5 / NUM_DAYS
           AVG_LEGUME = FG6 / NUM_DAYS
        ELSE
           AVG_FG1 = 0.0
           AVG_FG2 = 0.0
           AVG_FG3 = 0.0
           AVG_FG4 = 0.0
           AVG_FG5 = 0.0
           AVG_LEGUME = 0.0
           WRITE(6,*) ' -- ERROR: FOUND NUMDAYS = 0 FOR HHID_15 = '
     &                ,HHID_15
           WRITE(6,*) ' '
        END IF

        IF(KFREQ .GT. 0) THEN
           WRITE(PRFILE,2010)
     &     FG1
     &     ,FG2
     &     ,FG3
     &     ,FG4
     &     ,FG5
     &     ,FG6
     &     ,AVG_FG1
     &     ,AVG_FG2                            44
```

```
      &     ,AVG_FG3
      &     ,AVG_FG4
      &     ,AVG_FG5
      &     ,AVG_LEGUME
      &     ,NUM_DAYS
      &     ,SPLIT_CNT
      &     ,AGG_CNT
2010  FORMAT(
      &    / '   FG1 = ',F6.3
      &    ,'   FG2 = ',F6.3
      &    ,'   FG3 = ',F6.3
      &    ,'   FG4 = ',F6.3
      &    ,'   FG5 = ',F6.3
      &    ,'   FG6 = ',F6.3
      &    /,'  AVG_FG1 = ',F6.3
      &    ,'  AVG_FG2 = ',F6.3
      &    ,'  AVG_FG3 = ',F6.3
      &    ,'  AVG_FG4 = ',F6.3
      &    ,'  AVG_FG5 = ',F6.3
      &    ,'  AVG_LEGUME = ',F6.3
      &    /,'  NUM_DAYS = ',F6.3
      &    /,'  SPLIT_CNT = ',I3
      &    ,'  AGG_CNT = ',I3
      &    )
        END IF   ! kfreq

C-----------------------------------
C  SKIP OVER RECORDS WITH
C  INVALID DATA
C-----------------------------------
        IF(SEX .NE. 1 .AND. SEX .NE. 2) THEN
           INVALID_SEX_CNT = INVALID_SEX_CNT + 1
           IF(INVALID_SEX_CNT .LE. 5) THEN
              WRITE(PRFILE,*) ' ***** ERROR: INVALID SEX CODE *****'
              KFREQ = 9
           END IF
           GO TO 900
        END IF
        ISEX = SEX

        IF(AGE .GT. 98) THEN
           VERY_OLD_AGE_CNT = VERY_OLD_AGE_CNT + 1
           IF(VERY_OLD_AGE_CNT .LE. 5) THEN
              WRITE(PRFILE,*) ' ***** WARNING: VERY OLD AGE CODE *****'
              KFREQ = 9
           END IF
        END IF

        IF(AGE .GE. 51) THEN
           IAGE = 8
        ELSE IF(AGE .GE. 25) THEN
           IAGE = 7
        ELSE IF(AGE .GE. 19) THEN
           IAGE = 6
        ELSE IF(AGE .GE. 15) THEN
           IAGE = 5
        ELSE IF(AGE .GE. 11) THEN
           IAGE = 4
        ELSE IF(AGE .GE. 7) THEN
           IAGE = 3
        ELSE IF(AGE .GE. 4) THEN
           IAGE = 2
        ELSE IF(AGE .GE. 1) THEN
           IAGE = 1
        ELSE
           GO TO 900
        END IF

        IF(KFREQ .GT. 0) THEN
           WRITE(PRFILE,2110) ISEX,IAGE
2110       FORMAT('  ISEX = ',I2,'  IAGE = ',I2)
        END IF

C------ do the legume-meat before veggies if legumes gt 0
C------ give the meat excess to veggies
        IF(AVG_LEGUME .GT. 0.0) THEN
```
45

```
          MAXPOINT = MAX_POINT(ISEX,IAGE,5)
          MINPOINT = MIN_POINT(ISEX,IAGE,5)
          IF(AVG_FG5 .LT. MAXPOINT) THEN       ! if more meat servings needed for max
             AVG_FG5 = AVG_FG5 + AVG_LEGUME    ! add legumes to meat
             IF(AVG_FG5 .GT. MAXPOINT) THEN    ! if now excess meat servings
                AVG_FG2 = AVG_FG2 + (AVG_FG5 - MAXPOINT) ! move that to veggies
                AVG_FG5 = MAXPOINT                       ! reset meat to max
                LEGUME_CNT = LEGUME_CNT + 1
                IF(LEGUME_CNT .LE. 1) THEN
                   WRITE(PRFILE,*) ' ---- FOUND LEGUME DISTRIBUTION:'
                   KFREQ = 9
                END IF
             END IF
C         IF(AVG_FG2 .LT. MAXPOINT) THEN       ! if more veg servings needed for max
C            AVG_FG2 = AVG_FG2 + AVG_LEGUME    ! add legumes to veggies
C            IF(AVG_FG2 .GT. MAXPOINT) THEN    ! if now excess veggies servings
C               AVG_FG5 = AVG_FG5 + (AVG_FG2 - MAXPOINT) ! move that to meats
C               AVG_FG2 = MAXPOINT                       ! reset veggies to max
C               LEGUME_CNT = LEGUME_CNT + 1
C               IF(LEGUME_CNT .LE. 1) THEN
C                  WRITE(PRFILE,*) ' ---- FOUND LEGUME DISTRIBUTION:'
C                  KFREQ = 9
C               END IF
C            END IF
             IF(KFREQ .GT. 0) THEN
                WRITE(PRFILE,2125) AVG_FG2,AVG_FG5
2125            FORMAT('      legumes distributed: '
     &                ,'  new AVG_FG2 = ',F6.3
     &                ,'  new AVG_FG5 = ',F6.3)
             END IF
          END IF
       END IF

C------ the grain component
       MAXPOINT = MAX_POINT(ISEX,IAGE,1)
       MINPOINT = MIN_POINT(ISEX,IAGE,1)

       IF(AVG_FG1 .GE. MAXPOINT) THEN
          COMP01 = 10.0
       ELSE IF(AVG_FG1 .LT. MINPOINT) THEN
          COMP01 = 0.0
       ELSE
          COMP01 = 10.0 * (AVG_FG1 - MINPOINT) / (MAXPOINT - MINPOINT)
       END IF
       IF(KFREQ .GT. 0) THEN
          WRITE(PRFILE,2120) MAXPOINT,MINPOINT,AVG_FG1,COMP01
2120      FORMAT(' FOR GRAIN:   MAXPOINT = ',F6.2,'  MINPOINT = ',F6.2
     &          ,'  AVG_FG1 = ',F6.3,'  COMP01 = ',F7.2)
       END IF

C------ the veggie component
       MAXPOINT = MAX_POINT(ISEX,IAGE,2)
       MINPOINT = MIN_POINT(ISEX,IAGE,2)
       IF(AVG_FG2 .GE. MAXPOINT) THEN
          COMP02 = 10.0
       ELSE IF(AVG_FG2 .LT. MINPOINT) THEN
          COMP02 = 0.0
       ELSE
          COMP02 = 10.0 * (AVG_FG2 - MINPOINT) / (MAXPOINT - MINPOINT)
       END IF
       IF(KFREQ .GT. 0) THEN
          WRITE(PRFILE,2130) MAXPOINT,MINPOINT,AVG_FG2,COMP02
2130      FORMAT(' FOR VEGGIES: MAXPOINT = ',F6.2,'  MINPOINT = ',F6.2
     &          ,'  AVG_FG2 = ',F6.3,'  COMP02 = ',F7.2)
       END IF

C------ the fruit component
       MAXPOINT = MAX_POINT(ISEX,IAGE,3)
       MINPOINT = MIN_POINT(ISEX,IAGE,3)
       IF(AVG_FG3 .GE. MAXPOINT) THEN
          COMP03 = 10.0
       ELSE IF(AVG_FG3 .LT. MINPOINT) THEN
          COMP03 = 0.0
       ELSE
          COMP03 = 10.0 * (AVG_FG3 - MINPOINT) / (MAXPOINT - MINPOINT)
       END IF
```

46

```
      IF(KFREQ .GT. 0) THEN
         WRITE(PRFILE,2135) MAXPOINT,MINPOINT,AVG_FG3,COMP03
2135     FORMAT('  FOR FRUIT:   MAXPOINT = ',F6.2,'  MINPOINT = ',F6.2
     &          ,'  AVG_FG3 = ',F6.3,'  COMP03 = ',F7.2)
      END IF

C------ the dairy component
      MAXPOINT = MAX_POINT(ISEX,IAGE,4)
      MINPOINT = MIN_POINT(ISEX,IAGE,4)
      IF(AVG_FG4 .GE. MAXPOINT) THEN
         COMP04 = 10.0
      ELSE IF(AVG_FG4 .LT. MINPOINT) THEN
         COMP04 = 0.0
      ELSE
         COMP04 = 10.0 * (AVG_FG4 - MINPOINT) / (MAXPOINT - MINPOINT)
      END IF
      IF(KFREQ .GT. 0) THEN
         WRITE(PRFILE,2140) MAXPOINT,MINPOINT,AVG_FG4,COMP04
2140     FORMAT('  FOR DAIRY:   MAXPOINT = ',F6.2,'  MINPOINT = ',F6.2
     &          ,'  AVG_FG4 = ',F6.3,'  COMP04 = ',F7.2)
      END IF

C------ the meat component after legumes have been added
      MAXPOINT = MAX_POINT(ISEX,IAGE,5)
      MINPOINT = MIN_POINT(ISEX,IAGE,5)

      IF(AVG_FG5 .GE. MAXPOINT) THEN
         COMP05 = 10.0
      ELSE IF(AVG_FG5 .LT. MINPOINT) THEN
         COMP05 = 0.0
      ELSE
         COMP05 = 10.0 * (AVG_FG5 - MINPOINT) / (MAXPOINT - MINPOINT)
      END IF
      IF(KFREQ .GT. 0) THEN
         WRITE(PRFILE,2150) MAXPOINT,MINPOINT,AVG_FG5,COMP05
2150     FORMAT('  FOR MEATS:   MAXPOINT = ',F6.2,'  MINPOINT = ',F6.2
     &          ,'  AVG_FG5 = ',F6.3,'  COMP05 = ',F7.2)
      END IF

C------ the fat components
      IF(TOTNUT2 .GT. 0.0) THEN
         ALL_FAT_RATIO = (9.0*TOTNUT4) / TOTNUT2
         SATURATED_FAT_RATIO = (9.0*TOTNUT5) / TOTNUT2
      ELSE
         WRITE(PRFILE,*) ' **ERROR: TOTNUT2=0 FOR: PERSON#',TYPE_20_CNT
         KFREQ = 1
         ALL_FAT_RATIO = 0.0
         SATURATED_FAT_RATIO = 0.0
      END IF

C------ the all fat components
      MAXPOINT = MAX_POINT(ISEX,IAGE,6)
      MINPOINT = MIN_POINT(ISEX,IAGE,6)
      IF(ALL_FAT_RATIO .LT. MAXPOINT) THEN
         COMP06 = 10.0
      ELSE IF(ALL_FAT_RATIO .GE. MINPOINT) THEN
         COMP06 = 0.0
      ELSE
         COMP06 = 10.0 * (MINPOINT - ALL_FAT_RATIO)
     &               / (MINPOINT - MAXPOINT)
      END IF
      IF(KFREQ .GT. 0) THEN
         WRITE(PRFILE,2160) MAXPOINT,MINPOINT,ALL_FAT_RATIO,COMP06
2160     FORMAT('  FOR ALL FAT: MAXPOINT = ',F6.2,'  MINPOINT = ',F6.2
     &          ,'  ALL_FAT_RATIO = ',F6.3,'  COMP06 = ',F6.2)
      END IF


C------ the saturated fat components
      MAXPOINT = MAX_POINT(ISEX,IAGE,7)
      MINPOINT = MIN_POINT(ISEX,IAGE,7)
      IF(SATURATED_FAT_RATIO .LT. MAXPOINT) THEN
         COMP07 = 10.0
      ELSE IF(SATURATED_FAT_RATIO .GE. MINPOINT) THEN
         COMP07 = 0.0
      ELSE
```

47

```
        COMP07 = 10.0 * (MINPOINT - SATURATED_FAT_RATIO)
     &             / (MINPOINT - MAXPOINT)
      END IF
      IF(KFREQ .GT. 0) THEN
         WRITE(PRFILE,2170) MAXPOINT,MINPOINT,SATURATED_FAT_RATIO,COMP07
2170     FORMAT('  FOR SAT FAT: MAXPOINT = ',F6.2,'  MINPOINT = ',F6.2
     &             ,'  SAT_FAT_RATIO = ',F6.3,'  COMP07 = ',F6.2)
      END IF




C------ the cholesterol component
      MAXPOINT = MAX_POINT(ISEX,IAGE,8)
      MINPOINT = MIN_POINT(ISEX,IAGE,8)
      IF(TOTNUT8 .LT. MAXPOINT) THEN
         COMP08 = 10.0
      ELSE IF(TOTNUT8 .GE. MINPOINT) THEN
         COMP08 = 0.0
      ELSE
         COMP08 = 10.0 * (MINPOINT - TOTNUT8)
     &             / (MINPOINT - MAXPOINT)
      END IF
      IF(KFREQ .GT. 0) THEN
         WRITE(PRFILE,2180) MAXPOINT,MINPOINT,TOTNUT8,COMP08
2180     FORMAT('  FOR CHOLEST: MAXPOINT = ',F6.0,'  MINPOINT = ',F6.0
     &             ,'  TOTNUT8       = ',F6.0,'  COMP08 = ',F6.2)
      END IF




C------ the SODIUM component
      MAXPOINT = MAX_POINT(ISEX,IAGE,9)
      MINPOINT = MIN_POINT(ISEX,IAGE,9)
      IF(TOTNUT29 .LT. MAXPOINT) THEN
         COMP09 = 10.0
      ELSE IF(TOTNUT29 .GE. MINPOINT) THEN
         COMP09 = 0.0
      ELSE
         COMP09 = 10.0 * (MINPOINT - TOTNUT29)
     &             / (MINPOINT - MAXPOINT)
      END IF
      IF(KFREQ .GT. 0) THEN
         WRITE(PRFILE,2190) MAXPOINT,MINPOINT,TOTNUT29,COMP09
2190     FORMAT('  FOR SODIUM:  MAXPOINT = ',F6.0,'  MINPOINT = ',F6.0
     &             ,'  TOTNUT29       = ',F6.0,'  COMP09 = ',F6.2)
      END IF

C---- count food portions  > 0.5 in for the variety index
      VARIETY_CNT = 0
      DO I = 1,AGG_CNT
         IF(AGG_PORTION(I) .GT. 0.50) VARIETY_CNT = VARIETY_CNT + 1
      END DO



C------ the VARIETY component
      MAXPOINT = MAX_POINT(ISEX,IAGE,10)
      MINPOINT = MIN_POINT(ISEX,IAGE,10)
      IF(VARIETY_CNT .GT. MAXPOINT) THEN
         COMP10 = 10.0
      ELSE IF(VARIETY_CNT .LE. MINPOINT) THEN
         COMP10 = 0.0
      ELSE
         COMP10 = 10.0 * (VARIETY_CNT - MINPOINT)
     &             / (MAXPOINT - MINPOINT)
      END IF

      IF(KFREQ .GT. 0) THEN
         WRITE(PRFILE,2200) MAXPOINT,MINPOINT
     &          ,SPLIT_CNT
     &          ,AGG_CNT
     &          ,VARIETY_CNT
     &          ,COMP10
2200     FORMAT('  FOR Variety: MAXPOINT = ',F7.2
     &             ,'  MINPOINT = ',F7.2
     &             /,'  SPLIT_CNT =',I3
     &             ,'  AGG_CNT =',I3
     &             ,'  VARIETY_CNT =',I3
```

```
      &            ,'  COMP10        =',F6.2
      &            )
       END IF

       FOOD_INDEX1 =
      &          COMP01
      &        + COMP02
      &        + COMP03
      &        + COMP04
      &        + COMP05
      &        + COMP06
      &        + COMP07
      &        + COMP08
      &        + COMP09
      &        + COMP10

       IF(KFREQ .GT. 0) THEN
          WRITE(PRFILE,2210)
      &          COMP01
      &         ,COMP02
      &         ,COMP03
      &         ,COMP04
      &         ,COMP05
      &         ,COMP06
      &         ,COMP07
      &         ,COMP08
      &         ,COMP09
      &         ,COMP10
      &         ,RDA1
      &         ,FOOD_INDEX1
2210    FORMAT(
      &      / '  COMP01 =',F6.3
      &      ,'  COMP02 =',F6.3
      &      ,'  COMP03 =',F6.3
      &      ,'  COMP04 =',F6.3
      &      ,'  COMP05 =',F6.3
      &      /,'  COMP06 =',F6.3
      &      ,'  COMP07 =',F6.3
      &      ,'  COMP08 =',F6.3
      &      ,'  COMP09 =',F6.3
      &      ,'  COMP10 =',F6.3
      &      /,'  RDA1_FOOD_RATIO =',F8.3
      &      /,'  FOOD_INDEX       =',F8.3
      &      )
       END IF


900    CONTINUE

       RETURN
       END

       SUBROUTINE BINARY_SEARCH_INT4(KEY,PICK,LOW,HIGH,KFREQ)
       IMPLICIT NONE
C***********************************************************
C--- BINARY SEARCH ON AN ARRAY OF INTEGER*4               *
C***********************************************************
       INTEGER*4 PICK,INTERV,HI,LO
       INTEGER*4 KFREQ
       INTEGER*4 LOW
       INTEGER*4 HIGH
       LOGICAL DBUG
       INTEGER*4 KEY
       INCLUDE 'F7_CODE.INC'
C
C---- INTERVAL GT 1: COMPUTE NEW HI AND LO VALUES UNTIL INTERVAL <= 1
C---- THEN CHECK TO SEE WHICH OF THE TWO CHIOCES PICKES
C---- THIS ROUTINE ALWAYS EXPECTS TO FIND A PICK
       HI = HIGH
       LO = LOW
       DBUG = .FALSE.
  100  CONTINUE
       INTERV = HI - LO
       IF (INTERV .LE. 1) GOTO 200
          PICK = (LO + HI) / 2
```

49

```
          IF (KEY .LE. FOODCODE7(PICK)) THEN
              HI = PICK
          ELSE
              LO = PICK
          ENDIF
          GOTO 100

200   CONTINUE
C---- SEARCH INTERVAL DOWN TO 1 OR 0
C---- IF THE PICK IS NOT = TO THE KEY, THEN TRY THE NEXT SUBSCRIPT
C---- IN EITHER DIRECTION
      IF(FOODCODE7(PICK) .NE. KEY) THEN
          IF(FOODCODE7(PICK+1) .EQ. KEY) PICK = PICK + 1
          IF(FOODCODE7(PICK-1) .EQ. KEY) PICK = PICK - 1
      ENDIF

c---- there's a problem if the key is still not equal
      IF(FOODCODE7(PICK) .NE. KEY) THEN
          WRITE(8,2010) KEY
2010      FORMAT(' *** ERROR IN BINARY_SEARCH: NO PICK FOUND FOR:',I9)
          KFREQ = 9
          DBUG = .TRUE.
      ENDIF
      IF (DBUG) THEN
          WRITE(8,2000) KEY,HI,LO,PICK,
     &                  FOODCODE7(PICK),FOODCODE7(LO),FOODCODE7(HI)
2000      FORMAT(' RESULTS OF BINARY_SEARCH_INT4 '/' KEY = ',I10,
     1            '   HI = ',I5,' LO = ',I5,
     2          / ' PICK = ',I5,'  FOODCODE7(PICK) = ',I10,
     3            ' FOODCODE7(LO) = ',I10,'  FOODCODE7(HI) = ',I10)
      END IF

      RETURN
      END


      SUBROUTINE ZERO_INDEX
C**********************************************************
C* ASSIGN ALL INDEX-RELATED VARIABLES TO ZERO!
C**********************************************************
      IMPLICIT NONE

      INCLUDE 'INDEX.INC'

      FG1 = 0.0
      FG2 = 0.0
      FG3 = 0.0
      FG4 = 0.0
      FG5 = 0.0
      FG6 = 0.0
      AVG_FG1 = 0.0
      AVG_FG2 = 0.0
      AVG_FG3 = 0.0
      AVG_FG4 = 0.0
      AVG_FG5 = 0.0
      COMP01  = 0.0
      COMP02  = 0.0
      COMP03  = 0.0
      COMP04  = 0.0
      COMP05  = 0.0
      COMP06  = 0.0
      COMP07  = 0.0
      COMP08  = 0.0
      COMP09  = 0.0
      COMP10  = 0.0
      ALL_FAT_RATIO = 0.0
      SATURATED_FAT_RATIO = 0.0
      FOOD_INDEX1 = 0.0

      RETURN
      END


      SUBROUTINE WRITE_PERSON
C**********************************************************
C* WRITE OUT PERSON RECORDS
C**********************************************************
```

50

```
       IMPLICIT NONE

       INCLUDE 'WORKCSF.INC'
       INCLUDE 'INDEX.INC'

       TOT_COMP01 = TOT_COMP01 + COMP01 * WGT_3DAY
       TOT_COMP02 = TOT_COMP02 + COMP02 * WGT_3DAY
       TOT_COMP03 = TOT_COMP03 + COMP03 * WGT_3DAY
       TOT_COMP04 = TOT_COMP04 + COMP04 * WGT_3DAY
       TOT_COMP05 = TOT_COMP05 + COMP05 * WGT_3DAY
       TOT_COMP06 = TOT_COMP06 + COMP06 * WGT_3DAY
       TOT_COMP07 = TOT_COMP07 + COMP07 * WGT_3DAY
       TOT_COMP08 = TOT_COMP08 + COMP08 * WGT_3DAY
       TOT_COMP09 = TOT_COMP09 + COMP09 * WGT_3DAY
       TOT_COMP10 = TOT_COMP10 + COMP10 * WGT_3DAY

       TOT_WGT = TOT_WGT + WGT_3DAY


       WRITE(10)
     &      RT_15
     &     ,YEAR_15
     &     ,SAMPLE_15
     &     ,HHID_15
     &     ,LGRADE_15
     &     ,FOODSEC
     &     ,SURPLUS
     &     ,RT_20
     &     ,YEAR_20
     &     ,SAMPLE_20
     &     ,HHID_20
     &     ,R_NUM_20
     &     ,GEOG
     &     ,URB
     &     ,HHSZ
     &     ,INCOME
     &     ,PCTPOV
     &     ,H_RCVFS
     &     ,DHKHH
     &     ,CHILDREN
     &     ,HEADSTAT
     &     ,R_RCVFS
     &     ,AGE
     &     ,SEX
     &     ,PL_STAT
     &     ,RACE_20
     &     ,ORIGIN
     &     ,NUMDAYS
     &     ,R_WIC
     &     ,R_EMP
     &     ,R_LGRADE
     &     ,R_CNTL
     &     ,R_FREEL
     &     ,HLTHDIET
     &     ,SPECDIET
     &     ,DT_CALOR
     &     ,DS_CALOR
     &     ,DT_FAT
     &     ,DS_FAT
     &     ,DT_SALT
     &     ,DS_SALT
     &     ,DT_SUGAR
     &     ,DS_SUGAR
     &     ,DT_DIAB
     &     ,DS_DIAB
     &     ,DT_LFIB
     &     ,DS_LFIB
     &     ,DT_HFIB
     &     ,DS_HFIB
     &     ,DT_OTHER
     &     ,DS_OTHER
     &     ,VEGET
     &     ,R_WGT
     &     ,R_HGT
     &     ,BMI
     &     ,LEISURE
```

51

```
&       ,ACTIVE_L
&       ,D1_MNTH
&       ,D1_DATE
&       ,D1_YEAR
&       ,D1_DAY
&       ,D1_NREC
&       ,D1_FEW
&       ,D1_AMTUS
&       ,D1_RESUS
&       ,D2_MNTH
&       ,D2_DATE
&       ,D2_YEAR
&       ,D2_DAY
&       ,D2_NREC
&       ,D2_FEW
&       ,D2_AMTUS
&       ,D2_RESUS
&       ,D3_MNTH
&       ,D3_DATE
&       ,D3_YEAR
&       ,D3_DAY
&       ,D3_NREC
&       ,D3_FEW
&       ,D3_AMTUS
&       ,D3_RESUS
&       ,WGT_3DAY
&       ,RT_50
&       ,YEAR_50
&       ,SAMPLE_50
&       ,HHID_50
&       ,R_NUM_50
&       ,RACE_50
&       ,F1A
&       ,F1B
&       ,F1C
&       ,F1D
&       ,F1E
&       ,F4A
&       ,F4B
&       ,F4C1
&       ,F4C2
&       ,F4C3
&       ,F4D
&       ,F4E1
&       ,F4E2
&       ,F4F
&       ,F4G
&       ,F4H
&       ,F4I
&       ,F4J
&       ,F4K
&       ,F4L
&       ,F4M
&       ,F4N
&       ,F4O
&       ,RT_40
&       ,YEAR_40
&       ,SAMPLE_40
&       ,HHID_40
&       ,R_NUM_40
&       ,RACE_40
&       ,DAYCODE_40
&       ,BMILK
&       ,RDA1
&       ,RDA2
&       ,RDA3
&       ,RDA4
&       ,RDA5
&       ,RDA6
&       ,RDA7
&       ,RDA8
&       ,RDA9
&       ,RDA10
&       ,RDA11
&       ,RDA12
&       ,RDA13
```

52

```
      &    ,RDA14
      &    ,RDA15
      &    ,RDA16
      &    ,RDA17
      &    ,TOTNUT1
      &    ,TOTNUT2
      &    ,TOTNUT3
      &    ,TOTNUT4
      &    ,TOTNUT5
      &    ,TOTNUT6
      &    ,TOTNUT7
      &    ,TOTNUT8
      &    ,TOTNUT9
      &    ,TOTNUT10
      &    ,TOTNUT11
      &    ,TOTNUT29
      &    ,TYPE_30_CNT
      &    ,FG1
      &    ,FG2
      &    ,FG3
      &    ,FG4
      &    ,FG5
      &    ,FG6
      &    ,AVG_FG1
      &    ,AVG_FG2
      &    ,AVG_FG3
      &    ,AVG_FG4
      &    ,AVG_FG5
      &    ,AVG_LEGUME
      &    ,COMP01
      &    ,COMP02
      &    ,COMP03
      &    ,COMP04
      &    ,COMP05
      &    ,COMP06
      &    ,COMP07
      &    ,COMP08
      &    ,COMP09
      &    ,COMP10
      &    ,SPLIT_CNT
      &    ,AGG_CNT
      &    ,ALL_FAT_RATIO
      &    ,SATURATED_FAT_RATIO
      &    ,FOOD_INDEX1
       OUT_CNT = OUT_CNT + 1

       RETURN
       END


       SUBROUTINE TIMER(KEOF,PRFILE)
C*****************************************************************
C* TIME: ROUTINE USED TO DO TIMING
C*****************************************************************

       IMPLICIT NONE
       INTEGER*4 KEOF
       INTEGER*4 PRFILE

       INTEGER*2 HOUR1,MINUTE1,SECOND1,SEC100TH1
       INTEGER*2 HOUR2,MINUTE2,SECOND2,SEC100TH2
       INTEGER*2 TOT_HOURS,TOT_MINUTES,TOT_SECONDS

C****************************
C-      CHECK FOR KEOF   *
C****************************
       IF(KEOF .EQ. 2) THEN
          GO TO 200
       ELSE IF(KEOF .EQ. 1) THEN
          GO TO 100
       ELSE
          GO TO 300
       END IF

100    CONTINUE
```

53

```
C***********************
C          KEOF = 1      *
C***********************
      CALL GETTIM(HOUR1,MINUTE1,SECOND1,SEC100TH1)
      GO TO 900



200   CONTINUE
C***********************
C          KEOF = 2      *
C***********************
      CALL GETTIM(HOUR2,MINUTE2,SECOND2,SEC100TH2)
      TOT_HOURS = HOUR2 - HOUR1
      IF(TOT_HOURS .LT. 0) TOT_HOURS = TOT_HOURS + 24
      TOT_MINUTES = MINUTE2 - MINUTE1
      IF(TOT_MINUTES .LT. 0) THEN
        TOT_MINUTES = TOT_MINUTES + 60
        TOT_HOURS = TOT_HOURS - 1
      ENDIF
      TOT_SECONDS = SECOND2 - SECOND1
      IF(TOT_SECONDS .LT. 0) THEN
        TOT_SECONDS = TOT_SECONDS + 60
        TOT_MINUTES = TOT_MINUTES - 1
      ENDIF
      WRITE(6,2120) TOT_HOURS,TOT_MINUTES,TOT_SECONDS
2120  FORMAT(' ELAPSED TIME: ',I2,':',I2.2,':',I2.2)
      GO TO 900

300   CONTINUE
C***********************
C          KEOF = 3      *
C***********************
C---- GET EOF SYSTEM TIME
      CALL GETTIM(HOUR2,MINUTE2,SECOND2,SEC100TH2)
      TOT_HOURS = HOUR2 - HOUR1
      IF(TOT_HOURS .LT. 0) TOT_HOURS = TOT_HOURS + 24
      TOT_MINUTES = MINUTE2 - MINUTE1
      IF(TOT_MINUTES .LT. 0) THEN
        TOT_MINUTES = TOT_MINUTES + 60
        TOT_HOURS = TOT_HOURS - 1
      ENDIF
      TOT_SECONDS = SECOND2 - SECOND1
      IF(TOT_SECONDS .LT. 0) THEN
        TOT_SECONDS = TOT_SECONDS + 60
        TOT_MINUTES = TOT_MINUTES - 1
      ENDIF
      WRITE(PRFILE,9500) TOT_HOURS,TOT_MINUTES,TOT_SECONDS
9500  FORMAT(/' ELAPSED TIME: HOURS:',I2,'  MINUTES: ',I2,
     &        '  SECONDS: ',I2)
      WRITE(6,9510) TOT_HOURS,TOT_MINUTES,TOT_SECONDS
9510  FORMAT(' ELAPSED TIME : HOURS:',I2,'  MINUTES: ',I2,
     &        '  SECONDS: ',I2)

900   CONTINUE
      RETURN
      END
      SUBROUTINE GETTIM(HOUR,MINUTE,SECOND,SEC100TH)
C*****************************************************************
C---- GET THE SYSTEM TIME FOR THE LAHEY FORTRAN VERSION.  NOT NEEDED IN
C---- MS FORTRAN V5.0
C *****************************************************************
      CHARACTER*12 C_TIME
      INTEGER*2 HOUR,MINUTE,SECOND,SEC100TH

      CALL TIME(C_TIME)
      READ(C_TIME,1000) HOUR,MINUTE,SECOND,SEC100TH
1000  FORMAT(I2,1X,I2,1X,I2,1X,I2)
      RETURN
      END
```

APPENDIX D

EXAMPLES OF SAS PROGRAMS WHICH PRODUCE TABLES FROM
THE ANALYSIS FILE

```
* Program: tab2_89.sas;
* Author:  Karen Pence, 12/19/94;
* Data:    CSFINDEX.SSD (1989 data);
* Object:  show distribution of Healthy Eating Index for the 89 file,
*          using the new variety file;
* --------------------------------------------------------------------;

libname in 'c:\hdi89kmp';
options nocenter ls=132 ps=80;
* options obs = 100;
options formchar='83C4DAC2BFC3C584C0C1D9'X;

proc format;
  value indexfmt
   0-30 = '<= 30'
  30<-40 = '31-40'
  40<-50 = '41-50'
  50<-60 = '51-60'
  60<-70 = '61-70'
  70<-80 = '71-80'
  80<-90 = '81-90'
  90<-HIGH = '> 90';
run;

proc tabulate data = in.csfindex;
  class hd_index;
  format hd_index indexfmt.;
  freq wgt_3day;
  table hd_index, n pctn<hd_index>;
  title 'Table 2: Weighted 1989 data, using new TAS variety file';
run;

proc tabulate data = in.csfindex;
  class hd_index;
  format hd_index indexfmt.;
  table hd_index, n pctn<hd_index>;
  title 'Table 2: Unweighted 1989 data, using new TAS variety file';
run;

proc means data = in.csfindex;
  var hd_index;
  weight wgt_3day;
  title 'Weighted Mean HEI value, 1989 data, using new TAS variety file';
run;

proc means data = in.csfindex;
  var hd_index;
  title 'Unweighted Mean HEI value, 1989 data, using new TAS variety file';
run;
```

57

Table 2: Weighted 1989 data, using new TAS variety file

	N	PCTN
Healthy eating index value		
<= 30	128.00	0.06
31-40	4783.00	2.06
41-50	26567.00	11.46
51-60	60391.00	26.05
61-70	64673.00	27.90
71-80	50077.00	21.60
81-90	23095.00	9.96
> 90	2119.00	0.91

Table 2: Unweighted 1989 data, using new TAS variety file
23, 1994   2

	N	PCTN
Healthy eating index value		
<= 30	4.00	0.10
31-40	109.00	2.73
41-50	499.00	12.48
51-60	1038.00	25.97
61-70	1152.00	28.82
71-80	816.00	20.42
81-90	338.00	8.46
> 90	41.00	1.03

Weighted Mean HEI value, 1989 data, using new TAS variety file

Analysis Variable : HD_INDEX Healthy eating index value

N	Mean	Std Dev	Minimum	Maximum
3997	63.9057583	93.6552858	22.2209435	97.5870590

Unweighted Mean HEI value, 1989 data, using new TAS variety file

Analysis Variable : HD_INDEX Healthy eating index value

N	Mean	Std Dev	Minimum	Maximum
3997	63.1218388	12.4371765	22.2209435	97.5870590

```
* Program: tablex.sas;
* Author:  Karen Pence, 5/10/94;
* Data:    CSFINDEX.SSD;
* Object:  creates a table that calculates the mean healthy diet index
           score for selected subgroups of the population.  Modified 6/8/94.;
* -------------------------------------------------------------------------;

libname in 'c:\hdi89';
*options obs = 100;
options ps = 80 ls = 132 nocenter;
options formchar='B3C4DAC2BFC3C5B4C0C1D9'X;

proc format;
  value sexfmt
  1 = 'Male'
  2 = 'Female';
run;

proc format;
  value agefmt
   2-4  = '2 - 4'
   5-14 = '5 - 14'
  15-39 = '15 - 39'
  40-64 = '40 - 64'
  65-HIGH = '65 +';
run;

proc format;
  value racefmt
  1 = 'White'
  2 = 'Black'
  3 = 'Asian / Pacific Islander'
  4 = 'Aleut, American Indian, Eskimo'
  5 = 'Other';
run;

proc format;
  value origfmt
  1 = 'Hispanic'
  2 = 'Non-Hispanic';
run;

proc format;
  value headfmt
  1 = 'Male & Female Head'
  2 = 'Female Head'
  3 = 'Male Head';
run;

proc format;
  value educfmt
  0 -  9 = '< 2 yrs HS'
 10 - 11 = '2 - 3 yrs HS'
      12 = '4 yrs HS'
 13 - 15 = 'Some college'
      16 = '4 yrs college'
      17 = '> 4 yrs college'
      99 = 'Indeterminable';
run;

proc format;
  value pctfmt
    0 -  50  = '  0-50 %'
   51 - 100  = ' 51-100 %'
  101 - 130  = '101-130 %'
  131 - 200  = '131-200 %'
  201 - 300  = '201-300 %'
  301 - HIGH = '301% +';
run;

proc format;
  value progfmt
  1 = 'Yes'
  2 = 'No'
  9 = 'No answer';
run;
```

59

```
proc format;
  value lunchfmt
  1 = 'Free lunch'
  2 = 'Reduced price'
  3 = 'Full price'
  0 = 'Not applicable';
run;

proc format;
  value secfmt
  1 = 'Enough/Desired'
  2 = 'Enough/non-desired'
  3 = 'Sometimes not enough'
  4 = 'Often not enough';
run;

proc tabulate data = in.csfindex format = 8.1;
  class sex age race_20 origin headstat lgrade15 pctpov h_rcvfs r_wic r_freel
        surplus fooddesc;
  var hd_index;
  table age race_20 origin headstat lgrade15 pctpov h_rcvfs r_wic r_freel
        surplus fooddesc all='Total', hd_index='Dietary Index Score'*(sex
        all='Total')*(n mean);
  weight wgt_3day;
  format sex sexfmt. age agefmt. race_20 racefmt. origin origfmt.
         headstat headfmt. lgrade15 educfmt. pctpov pctfmt.
         fooddesc secfmt. h_rcvfs r_wic surplus progfmt. r_freel lunchfmt.;
  where age gt 1;
  label race_20  = 'RACE'
        lgrade15 = 'EDUCATION OF HOUSEHOLD HEAD'
        pctpov   = 'POVERTY LINE PERCENTAGE'
        origin   = 'ETHNICITY'
        headstat = 'HEAD OF HOUSEHOLD'
        fooddesc = 'FOOD SECURITY'
        h_rcvfs  = 'FOOD STAMP PARTICIPANT'
        r_wic    = 'WIC PARTICIPANT'
        r_freel  = 'SCHOOL LUNCH PARTICIPANT'
        surplus  = 'USDA SURPLUS PARTICIPANT';
  title 'Table X - Mean Dietary Index Scores for Selected Population Subgroups';
  title2 'Weighted 1989 data';
run;
```

Table X - Mean Dietary Index Scores for Selected Population Subgroups

	Dietary Index Score					
	SEX				Total	
	Male		Female			
	N	MEAN	N	MEAN	N	MEAN
AGE						
2 - 4	122.0	70.3	117.0	73.3	239.0	71.8
5 - 14	368.0	66.0	330.0	66.2	698.0	66.1
15 - 39	648.0	58.7	795.0	62.3	1443.0	60.5
40 - 64	424.0	60.3	553.0	66.5	977.0	63.5
65 +	240.0	68.2	400.0	69.9	640.0	69.2
RACE						
White	1484.0	62.3	1759.0	66.3	3243.0	64.3
Black	222.0	58.0	319.0	60.7	541.0	59.5
Asian / Pacific Islander	36.0	70.3	40.0	71.9	76.0	70.9
Aleut, American Indian, Eskimo	17.0	49.1	21.0	58.2	38.0	54.2
Other	43.0	63.5	56.0	63.6	99.0	63.5
ETHNICITY						
Hispanic	166.0	61.0	240.0	64.2	406.0	62.7
Non-Hispanic	1636.0	62.1	1955.0	65.8	3591.0	63.9
HEAD OF HOUSEHOLD						
Male & Female Head	1367.0	62.5	1382.0	65.8	2749.0	64.1
Female Head	210.0	61.5	807.0	65.2	1017.0	64.5
Male Head	225.0	58.6	6.0	66.5	231.0	58.8
EDUCATION OF HOUSEHOLD HEAD						
< 2 yrs HS	226.0	58.4	350.0	62.5	576.0	60.9
2 - 3 yrs HS	147.0	56.7	251.0	62.0	398.0	59.7
4 yrs HS	651.0	60.4	754.0	64.2	1405.0	62.3
Some college	380.0	62.0	431.0	65.9	811.0	63.9
4 yrs college	157.0	63.7	168.0	68.0	325.0	65.8
> 4 yrs college	211.0	66.6	206.0	69.8	417.0	68.2
Indeterminable	30.0	65.6	35.0	68.3	65.0	67.1
POVERTY LINE PERCENTAGE						
0-50 %	134.0	60.6	186.0	58.6	320.0	59.5
51-100 %	364.0	59.1	547.0	62.5	911.0	61.2
101-130 %	190.0	60.7	285.0	63.1	475.0	62.0
131-200 %	277.0	59.6	308.0	64.2	585.0	62.0
201-300 %	300.0	61.0	319.0	65.7	619.0	63.3

Table X - Mean Dietary Index Scores for Selected Population Subgroups
Weighted 1989 data

	Dietary Index Score					
	SEX				Total	
	Male		Female			
	N	MEAN	N	MEAN	N	MEAN
POVERTY LINE PERCENTAGE						
301% +	537.0	63.5	550.0	67.4	1087.0	65.4
FOOD STAMP PARTICIPANT						
Yes	206.0	61.1	327.0	60.5	533.0	60.8
No	1596.0	62.0	1868.0	66.1	3464.0	64.1
WIC PARTICIPANT						
Yes	27.0	70.7	30.0	64.8	57.0	67.3
No	1748.0	61.9	2133.0	65.6	3881.0	63.8
8	.	.	1.0	74.9	1.0	74.9
No answer	27.0	61.8	31.0	67.1	58.0	65.0
SCHOOL LUNCH PARTICIPANT						
Not applicable	1483.0	61.7	1879.0	65.7	3362.0	63.8
Free lunch	114.0	66.0	125.0	63.8	239.0	64.8
Reduced price	46.0	64.4	28.0	61.8	74.0	63.4
Full price	158.0	62.4	159.0	65.3	317.0	63.8
8	1.0	75.3	.	.	1.0	75.3
9	.	.	4.0	69.4	4.0	69.4
USDA SURPLUS PARTICIPANT						
Yes	126.0	58.4	208.0	61.9	334.0	60.6
No	1670.0	62.0	1979.0	65.7	3649.0	63.9
No answer	6.0	69.4	8.0	75.4	14.0	73.4
FOOD SECURITY						
Enough/Desired	1166.0	62.7	1403.0	66.9	2569.0	64.8
Enough/non-desired	547.0	60.3	706.0	62.2	1253.0	61.3
Sometimes not enough	72.0	55.8	68.0	62.1	140.0	59.1
Often not enough	15.0	61.0	15.0	53.3	30.0	57.8
9	2.0	42.1	3.0	45.2	5.0	43.9
Total	1802.0	62.0	2195.0	65.6	3997.0	63.8

```
* Program: energalc.sas;
* Author:  Karen Pence, 9/29/94;
* Data:    1989 CSFINDEX;
* Object:  tabulate body mass index categories by energy intake categories;
* ----------------------------------------------------------------------;

libname in 'c:\hdi89';
*options obs = 100;
options ps = 80 ls = 150 nocenter;
options formchar='83C4DAC2BFC3C584C0C1D9'X;

proc format;
  value bmifmt
  15  - 20  = '15 - 20'
  20 <- 25  = '20 <- 25'
  25 <- 30  = '25 <- 30'
  30 <- HIGH = '> 30';
run;

proc format;
  value alcfmt
  0          = 'None'
  0  <- 2.0  = '0.0 <- 2.0'
  2.0 <- 4.0 = '2.0 <- 4.0'
  4.0 <- 6.0 = '4.0 <- 6.0'
  6.0 <- 8.0 = '6.0 <- 8.0'
  8.0 <- HIGH = ' > 8.0';
run;

data energy;
  set in.csfindex (keep = bmi wgt_3day sex age totnut11 totnut2);

  if bmi > 999 then bmi = .;
  alc_cal = ((totnut11 * 7)/totnut2)*100;
  if (sex = 1) and (18 <= age <= 74) then output;

run;

proc tabulate data = energy format = 8.2;
  class bmi alc_cal;
  table alc_cal all, bmi*(n*f=8.0 pctn<bmi all>) all='Total'*(n*f=8.0 pctn)/rts = 22;
  label bmi = 'Body Mass Index'
        alc_cal = '% of Calories from Alcohol';
  keylabel pctn = 'Pct';
  format bmi bmifmt. alc_cal alcfmt.;
  title 'Percentage of Calories from Alcohol versus Body Mass Index, Weighted 1989 Data';
  freq wgt_3day;
run;

proc tabulate data = energy format = 8.2;
  class bmi alc_cal;
  table alc_cal all, bmi*(n*f=8.0 pctn<bmi all>) all='Total'*(n*f=8.0 pctn)/rts = 22;
  label bmi = 'Body Mass Index'
        alc_cal = '% of Calories from Alcohol';
  keylabel pctn = 'Pct';
  format bmi bmifmt. alc_cal alcfmt.;
  title 'Percentage of Calories from Alcohol versus Body Mass Index, Unweighted 1989 Data';
run;

proc corr data = energy;
  var bmi alc_cal;
  weight wgt_3day;
  title 'Weighted correlation between ALC_CAL & BMI';
run;

proc corr data = energy;
  var bmi alc_cal;
  title 'Unweighted correlation between ALC_CAL & BMI';
run;
```

63

Percentage of Calories from Alcohol versus Body Mass Index, Weighted 1989 Data

	Body Mass Index								Total	
	15 - 20		20 <- 25		25 <- 30		> 30			
	N	Pct	N	Pct	N	Pct	N	Pct	N	Pct
% of Calories from Alcohol										
None	2315	4.46	19707	37.95	21741	41.86	8172	15.74	51935	65.18
0.0 <- 2.0	289	4.87	2702	45.57	1712	28.88	1226	20.68	5929	7.44
2.0 <- 4.0	132	1.93	4120	60.11	2033	29.66	569	8.30	6854	8.60
4.0 <- 6.0	10	0.22	1496	32.19	2098	45.15	1043	22.44	4647	5.83
6.0 <- 8.0	300	11.65	928	36.04	1281	49.75	66	2.56	2575	3.23
> 8.0	112	1.45	3218	41.55	3650	47.13	764	9.87	7744	9.72
ALL	3158	3.96	32171	40.37	32515	40.80	11840	14.86	79684	100.00

Percentage of Calories from Alcohol versus Body Mass Index, Unweighted 1989 Data

	Body Mass Index								Total	
	15 - 20		20 <- 25		25 <- 30		> 30			
	N	Pct	N	Pct	N	Pct	N	Pct	N	Pct
% of Calories from Alcohol										
None	39	4.83	325	40.27	320	39.65	123	15.24	807	70.05
0.0 <- 2.0	4	4.76	37	44.05	34	40.48	9	10.71	84	7.29
2.0 <- 4.0	3	3.53	44	51.76	28	32.94	10	11.76	85	7.38
4.0 <- 6.0	1	2.38	20	47.62	16	38.10	5	11.90	42	3.65
6.0 <- 8.0	2	6.06	15	45.45	14	42.42	2	6.06	33	2.86
> 8.0	3	2.97	48	47.52	45	44.55	5	4.95	101	8.77
ALL	52	4.51	489	42.45	457	39.67	154	13.37	1152	100.00

CORRELATION ANALYSIS

    2 'VAR' Variables: BMI   ALC_CAL

                                         Simple Statistics

Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
BMI	1152	26.028035	35.938087	2074018	15.060000	54.810001
ALC_CAL	1167	2.280319	40.405026	183586	0	48.604586

Pearson Correlation Coefficients / Prob > |R| under Ho: Rho=0 / Number of Observations / WEIGHT Var = WGT_3DAY

	BMI	ALC_CAL
BMI	1.00000	-0.02692
	0.0	0.3613
	1152	1152
ALC_CAL	-0.02692	1.00000
	0.3613	0.0
	1152	1167

Unweighted correlation between ALC_CAL & BMI

CORRELATION ANALYSIS

    2 'VAR' Variables: BMI   ALC_CAL

                                         Simple Statistics

Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
BMI	1152	25.732613	4.279802	29644	15.060000	54.810001
ALC_CAL	1167	2.012299	4.978938	2348.353295	0	48.604586

Pearson Correlation Coefficients / Prob > |R| under Ho: Rho=0 / Number of Observations

	BMI	ALC_CAL
BMI	1.00000	-0.05624
	0.0	0.0564
	1152	1152
ALC_CAL	-0.05624	1.00000
	0.0564	0.0
	1152	1167